Image reconstruction

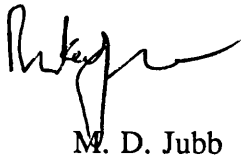
Jubb, M. D.

Award date:
1989

Awarding institution:
University of Bath

Image Reconstruction

submitted by M. D. Jubb
for the degree of PhD
of the University of Bath
1989



M. D. Jubb

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

Image reconstruction is used in a wide range of practical applications. The most general form of the problem is to obtain a reconstruction x from a set of data values y which convey imperfect information about the unknown true scene x^*. Statisticians have only recently become involved in image reconstruction, modelling the true scene as a spatial process. The prior model which is normally used is the Markov random field model which, loosely speaking, corresponds to a belief that neighbouring pixels (picture elements) in the true scene tend to be of the same or similar colour.

Chapter 2 of this thesis introduces the Markov random field model and the mathematical formulation of the problem that will be used. In Chapter 3 we describe several existing reconstruction techniques, and examine the basic properties of each alongside an example. Chapters 4 and 5 examine the possibilities of reconstructing the image on a different scale than that on which the data values were recorded. In Chapter 4 reconstructions are produced on different levels of the grid using an algorithm which enables the reconstruction of both large and small scale features of the image. In Chapter 5 reconstruction below the pixel level, known as subpixel refinement, is considered. The form of subpixel solutions is examined and a subpixel technique is proposed and implemented with good results.

If unsuitable parameters are used in the prior model then the resulting reconstruction may be oversmoothed. In Chapter 6 we propose a test which aims to detect oversmoothing in binary reconstructions.

In Chapter 7 we describe a grey level technique which combines a technique for finding exact binary MAP reconstructions with a method which divides a multi-level problem into several two level problems.

Acknowledgements

This research was carried out under the supervision of Dr. Christopher Jennison. I am extremely grateful for his guidance and encouragement during the course of this work.

I would also like to thank Julian Stander for his efficient and meticulous checking of the document, though of course any mistakes, omissions or obscurities remain entirely the responsibility of the author.

Thanks are also due to my colleagues Martyn, Shirley, Paul and Christine as well as other members of the school of Mathematical Sciences for their advice and suggestions.

Additional thanks and appreciation are due to Claire, Mags, Megan, other friends and members of my family for their support which came in a variety of forms.

Finally I must acknowledge the Science and Engineering Research Council for their generous financial support.

Table of Contents

Chapter 1: Introduction

Image reconstruction has a wide range of practical applications, for example medical imaging, computer vision, satellite imaging and optical astronomy. Some of the medical applications include thermal imaging, ultrasound, tomography, and nuclear medicine. In general, a set of data values relating to the true unknown image are observed, and a reconstruction of the image is obtained. The way in which the reconstruction is used depends on the application. In LANDSAT (land satellite) imaging, areas of the earths' surface are scanned and reconstructions are used to classify land use or rock type. In computer vision, image processing is used to classify shapes and identify features. Image processing may be used in medical imaging to establish the existence of unwelcome features such as growths. The sensing mechanisms which are used to collect the data vary greatly and the work in this thesis concentrates on one particular case, developing methods and approaches which generalise to other cases.

1.1 The problem

We concentrate on the case where the data have been directly observed. This case may be considered as a prototype for the indirectly observed case. Consider a two dimensional rectangular region partitioned into picture elements known as pixels. For each pixel a value or intensity, known as a record, is observed which conveys imperfect information relating to the true colouring of the pixel. The collection of all the individual records is known as the record for the scene. Given the complete record the problem is then to obtain a reconstruction which is as close as possible in some sense to the true scene. It is common to assume that the record for each pixel depends only on the colouring of that pixel, although there are techniques which are capable of handling any blurring or similar degradation.

1.2 The statistical approach

Statisticians have only recently become involved in image reconstruction, modelling the true scene as a spatial process. The prior model which is normally used is the Markov Random Field (MRF) model which, loosely speaking, corresponds to a belief that neighbouring pixels in the true scene tend to be of the same or similar colour. The key property of this model is that the distribution of the colouring of any pixel is conditionally independent of all other pixels, given

the colouring of its neighbours. The MRF model is described in Section 2.1.

The prior information is combined with the record and the problem reformulated as the minimisation of an objective function of two terms. The first of these measures the infidelity between the record and a given reconstruction; the second is a measure of the roughness of the reconstruction. In many of the applications that have been mentioned the size of the region will be large e.g. 256 by 256 pixels. Each of these pixels may be coloured with any of the grey levels available giving a very large number of different configurations. It is clear that the reconstruction necessitates the use of some kind of algorithm. The complexity of the reconstruction procedure will depend on the way in which the problem is formulated. Grenander (1983) and others have used a method of maximising marginal posterior modes (MPM) in which the most likely colouring of each pixel is sought. The majority of the statistical techniques search for the *maximum a posteriori* (MAP) estimate of the true scene as being that which has maximum probability in the Bayes' formulation. Geman and Geman (1984) propose a method which attempts to find the global minimum of the objective function (which corresponds to the MAP estimate) using the method of simulated annealing. The method of Iterated Conditional Modes (ICM) proposed by Besag (1986) searches for a local minimum of the objective function and convergence is faster. Both of these techniques are described in Chapter 3.

In addition to techniques which use MAP estimation there are several smoothing techniques which avoid the use of a formal prior, choosing instead to use filtering, maximum entropy, regularization or deconvolution. Such techniques are discussed by Hall and Titterington (1986).

1.3 Pixel size

The size of the pixels that are used, relative to the size of the features in the image, is often determined by the sensing mechanism. In some cases the pixels may be too large, so that two regions meet within a pixel; or the pixels may be unnecessarily small. In the application of SPET (single photon emission tomography) there are no predefined pixels, and the structure of the region may be freely chosen.

In Chapter 4 a technique which uses aggregation is developed, enabling the reconstruction of both the large and small scale features of the image. In Chapter

5 reconstruction below the pixel level, known as subpixel refinement, is considered. One example of the need for refinement is in the processing of LANDSAT data where each pixel may measure up to an acre in size. In this context Switzer and Venetoulias (1985) and others recognise the problem of classifying mixed pixels. Jennison (1986) used a modification of ICM which, for the two colour case, reconstructed the image on a finer grid than that on which the record had been collected. Each pixel was divided into 4 subpixel quarters and a separate colour allocated to each subpixel. In Chapter 5 of this thesis we extend the refinement process to the continuum, examine the theoretical properties of the solution and propose a technique for finding an approximation to this solution.

1.4 Other topics considered

As stated in Section 1.2, the reconstruction process can be regarded as the minimisation of an objective function of two terms. The trade off between the terms is controlled by a smoothing parameter which is built into the prior model. If this parameter is too large then the resulting reconstruction may be oversmoothed. In Chapter 6 we propose a test which aims to detect oversmoothing in binary reconstruction.

In Chapter 7 we describe a technique for reconstructing grey level images which combines a way of finding exact binary MAP reconstructions with a method which divides a multi-level problem into several two level problems.

Chapter 2: Notation and model

The notation that we will use follows closely the notation used by Besag (1986). We consider a two dimensional rectangular region S partitioned into pixels labelled $1,2,\ldots,n$. Each pixel has a true colour or grey level which is denoted by x_i^* for pixel i. The true colouring of the region is denoted by $x^*=\{x_i^*;i=1,\ldots,n\}$ and this is interpreted as the realisation of a random vector $X=\{X_1,X_2,\ldots,X_n\}$, where X_i denotes the colour of pixel i. The x_i^* are unobserved. Instead we observe the record at pixel i which conveys imperfect information about the true colouring x_i^* at pixel i. Throughout this thesis we shall assume that the records y_i are independently distributed as Gaussian with mean x_i^* and variance σ^2. The set of records is denoted by $y = \{y_i;\ i=1,\ldots,n\}$. A colouring of pixel i (not necessarily the true colouring, x_i^*) is denoted by x_i and a specific colouring of the whole region is denoted by $x = \{x_i;\ i=1,\ldots,n\}$. The sample space of x is Ω.

2.1 Setting up the Markov random field model for the true scene

A pixel j is defined to be a **neighbour** of pixel i ($\neq j$) if

$P(X_i=x_i \mid X_1=x_1,\ldots,X_{i-1}=x_{i-1},X_{i+1}=x_{i+1},\ldots,X_n=x_n)$ depends on x_j.

The set of neighbours of pixel i is denoted by ∂_i. Note that the above definition does not imply that pixels which are neighbours are geometrically close, although this would be the case in an imaging context.

A colouring of the region S, denoted by x is a realisation of a **Markov Random Field** (MRF) with respect to the particular neighbour relation if

(i) $P(X=x)>0$ for all x in Ω.

(ii) $P(X_r=x_r \mid X_t=x_t, t\neq r) = P(X_r=x_r \mid X_t=x_t, t\in\partial_r)$

Cross and Jain (1983) include a further constraint of homogeneity:

(iii) $P(X_r=x_r \mid X_t=x_t, t\in\partial_r)$ depends only on the configuration of neighbours and is homogeneous within the region.

The crucial condition is (ii), the Markov property. Given these conditional distributions, Besag (1974) demonstrates that the function $P(X=x)$, which we will write as $p(x)$, may be directly derived and gives the most general form of this distribution. In image processing applications a pairwise interactive MRF model is often used and the probability density is given by

$$p(x) \propto \exp\left\{\sum_{i=1}^{n} G_i(x_i) + \sum_{i=1}^{n}\sum_{j=1}^{n} G_{ij}(x_i,x_j)\right\}.$$

where $G_{ij} \equiv 0$ unless pixel i and j are neighbours and all other G functions are arbitrary. The choice of the G functions will depend on the application as well as the number of different colours and their ordering. Where the neighbourhood relation only exists between pixels that are geometrically close to one another the model is known as a *locally dependent* Markov Random Field Model. Besag (1986) gives examples of the G functions which might be used for different applications. The Ising model (Ising, 1925) is a simple case of this, which is used to model ferromagnetism. The model is given by

$$p(x) \propto \exp\left[\alpha\sum_{i=1}^{n} x_i + \beta\sum_{i=1}^{n}\sum_{j\in\partial_i} x_i x_j\right]$$

for some fixed parameters α and β which measure the external field and bonding strengths respectively. In this example x_i takes the value -1 or 1 only.

We now give the models for two simple cases used in image applications. The first of these is for a scene where each pixel takes one of c unordered colours and two pixels are considered neighbours if they are horizontally, vertically or diagonally adjacent. At pixel i, given the colouring of the rest of the scene, colour k occurs with probability

$$P(x_i=k \mid x_j, j\neq i) = \frac{e^{\beta A_i(k)}}{\sum_{l=1}^{c} e^{\beta A_i(l)}}$$

where $A_i(k)$ is the number of neighbours of pixel i which are coloured with colour k and β is a fixed parameter. The strength of association between neighbouring pixel depends on the choice of β. Choosing β large implies that there is strong neighbouring dependence while choosing $\beta=0$ implies that there is none. The above model has been expressed in terms of conditional probabilities but from these the probability density for the whole scene may be derived as

$$p(x) \propto \exp\left[\frac{\beta}{2}\sum_{i=1}^{n} A_i(x_i)\right]$$

Pixels are now defined to be *first order* neighbours if they are horizontally or vertically adjacent to each other and *second order* neighbours if they are diagonally adjacent. Another example is given for the two colour case, in which

all pixels are either black or white. The prior distribution for the true scene may be given by

$$p(x) \propto \exp[-\{\beta_1 Z_1(x)+\beta_2 Z_2(x)\}].$$

where $Z_1(x)$ is the number of discrepant first order pairs in the scene x, i.e., the number of pairs of first order neighbours which are of opposite colour, $Z_2(x)$ is the number of discrepant second order pairs and β_1 and β_2 are fixed positive constants.

2.2 The Bayes' formulation

There are several ways in which the prior model may be combined with the observed record. Bayes' Theorem can be used to combine the information supplied by the record with the prior model. Recall

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}.$$

The way in this which this formulation is used will depend on the qualities that are required of the reconstruction. Where we might wish to construct a crop inventory from satellite data, Besag points out that it may be pertinent to maximise the expected proportion of correctly classified pixels. This corresponds to the maximisation of $P(X_i=x_i^*|y)$, the marginal posterior probability of x_i at i given the record y. By Bayes' Theorem we may write

$$P(X_i=x_i^*|y) \propto \sum_{x\in S: x_i=\tilde{x}_i} l(y|x)p(x)$$

where $l(y|x)$ is the conditional likelihood of the observed record y, given the colouring x and $p(x)$ is the prior probability of x. This method is known as maximising marginal probabilities and its use in imaging is limited by the severe computational difficulties encountered in maximisation. This kind of estimation is done by simulating from the marginal distribution at each pixel. A more widely used formulation is that of MAP estimation. The *maximum a posteriori* (MAP) estimate of the true scene is the value of x which maximises $P(x|y)$, the conditional probability of x given the record y. Again the Bayes' formulation is used and we write

$$P(x|y) \propto l(y|x)p(x).$$

Chapter 3: Existing methods

In this chapter we describe a few of the statistical reconstruction techniques used for MAP estimation. Two of these, Simulated Annealing (Geman and Geman, 1984) and Iterated Conditional Modes (Besag, 1986) have proved fundamental in this area and have been fully implemented by the author; an example is used to demonstrate these methods. Other techniques have been developed which incorporate these methods and these shall be discussed later in this chapter. The intention of this chapter is to summarise these methods in a clear and concise manner, observing limitations, advantages and peculiarities while referring the reader to the works of the given authors for the detailed theoretical support.

3.1 The example

Figure 3.1 shows a 64 by 64 binary image.

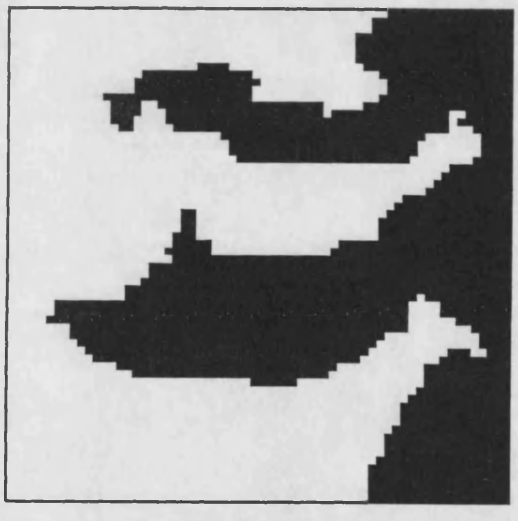


Figure 3.1

Each pixel is coloured black or white and the colour of pixel i is denoted by x_i^* which takes the value 0 for white and 1 for black. The records y_i are independently distributed as Gaussian with mean x_i^* and variance $\sigma^2=0.25$. Thus the likelihood of the record y given x is given by

$$l(y|x) = \prod_{i=1}^{n} f(y_i|x_i) = \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-x_i)^2\right]$$

The prior model which will be used in reconstruction is given by

$$p(x) \propto \exp[-\{\beta_1 Z_1(x)+\beta_2 Z_2(x)\}]$$

where $Z_1(x)$ is the number of discrepant first order pairs in the scene x, i.e., the number of pairs of first order neighbours which are of opposite colour, $Z_2(x)$ is the number of discrepant second order pairs and β_1 and β_2 are fixed positive constants. Hence the posterior probabilities may be calculated from

$$P(x|y) \propto l(y|x)p(x) \propto \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-x_i)^2\right]. \exp\left[-\{\beta_1 Z_1(x)+\beta_2 Z_2(x)\}\right]. \quad (3.1)$$

The maximisation of the posterior probability may be reformulated as the minimisation of the objective function

$$\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-x_i)^2 + \left[\beta_1 Z_1(x)+\beta_2 Z_2(x)\right] \quad (3.2)$$

over values of x.

The first term of the objective function is a measure of the infidelity between the record and the reconstruction and the second is a measure of the roughness of the reconstruction.

At this point some comment on the choice of the β values is required. There are several different methods for establishing a "good" value but a "try it and see" approach is commonly used. If large values of β_1 and β_2 are used then the objective function is dominated by the roughness term and the reconstruction which minimises (3.2) will be smooth. If small values of β_1 and β_2 are used then the reconstruction which minimises (3.2) will correspond closely to the record. Some inference as to the correct ratio of β_1 to β_2 has been made by Silverman, Jennison and Brown (1989) for the second order model. They have shown that using $\beta_2=\beta_1/\sqrt{2}$ minimises the rotational variance of the prior model with respect to the positioning of the pixel grid on a given scene. In Chapter 6 of this thesis we propose a technique for detecting oversmoothing in image reconstruction.

Before describing any techniques we define the closest mean classifier as being the image which contributes least to the first term of the objective function i.e. the image which has most fidelity with the record. Each pixel is coloured with the colour which has mean closest to its record e.g. if $y_i=0.4$ then pixel i is coloured white since

$$\left|0.4-0.0\right| < \left|0.4-1.0\right|.$$

In iterative methods it is usual to use the closest mean classifier as an initial estimate of the true scene.

3.2 Iterated conditional modes

Besag (1986) proposed the method of Iterated Conditional Modes (ICM) which finds a local maximum of the objective function (3.2) in the following way: The closest mean classifier is used as an initial estimate of the true scene. The region is scanned and at each pixel i say, the colouring x_i is chosen to minimise (3.2) over all possible x_i with all other pixel colourings fixed. This is equivalent to choosing x_i to minimise

$$\frac{1}{2\sigma^2}(y_i - x_i)^2 + \left[\beta_1 Z_1(x_i) + \beta_2 Z_2(x_i)\right]$$

where $Z_1(x_i)$ and $Z_2(x_i)$ are the number of first and second order neighbours of pixel i which are opposite colour to x_i. Inspection of (3.1) shows that this choice of x_i maximises

$$l(y_i \mid x_i) p(x_i \mid x_{S\setminus i}),$$

where $x_{S\setminus i}$ is the colouring of all pixels except the colouring at pixel i. This is equivalent to choosing x_i to maximise

$$l(y_i \mid x_i) p(x_i \mid x_j, j \in \partial_i).$$

It follows that this choice of x_i gives the most likely colouring at each pixel based only on its record and the colouring of its neighbours.

Each scan in which every pixel is considered for updating once is known as an iteration. The order in which the region is scanned is arbitrary if updating is synchronous since only the neighbour information from the last iteration will be used. It is, however, more convenient to scan the region in a raster fashion, storing only the current colouring of each pixel. When updating a pixel, the neighbours which are to the left or below the pixel have been more recently updated than those to the right or above the pixel. Besag (1986) notes that small directional effects may be introduced if the region is scanned in this way and suggests that these effects may be lessened if the raster is changed after each iteration. In the example that follows a straightforward raster scan is used for convenience and storage reasons. When a raster scan is used, the objective function must decrease or remain constant at each updating and because there are

a finite number of possible colourings a local minimum will be achieved in a finite number of iterations.

Besag (1986) uses $\beta_1=\beta_2=1.5$ for reconstruction but our experimental results suggest that this value is too high. This conclusion has also been reached by Ripley (1986). Another approach used by Besag was to increase the value of the smoothing parameter during reconstruction, settling at a value of 1.5. This encourages fidelity with the record in the early stages of reconstruction and avoids using what may be misleading spatial information early in the process. A similar approach has been adopted by Stander (1989) who has found that stopping the process much earlier, when $\beta=0.75$, has given superior results. Using the closest mean classifier as an initial estimate of the scene corresponds to using $\beta_1=\beta_2=0$ for the first iteration.

In the example we use a constant value of β for all iterations. Figure 3.2 shows the closest mean classifier for the record. In this initial estimate of the scene approximately 16% of the pixels are misclassified. Below each of the displayed reconstructions we give the number of misclassified pixels when compared to the true scene. Ripley (1986) warns of the dangers of using misclassification rates as a measure of goodness of fit and these figures should only be regarded as a rough guide to the quality of the reconstruction. Figure 3.3 shows the reconstruction obtained using ICM with $\beta_1=1$ and $\beta_2=0$. Setting $\beta_2=0$ is equivalent to using a first order model in which pixels are considered neighbours only if they are horizontally or vertically adjacent. A more interesting reconstruction is shown in Figure 3.4 which was obtained using ICM with $\beta_1=0$ and $\beta_2=1$. Reconstructing the image in this way corresponds to reconstructing two independent images, each containing 32×32 pixels diagonally connected (like the black squares on a chequerboard). The two separate regions cannot interact and this leads to the unusual reconstruction shown in Figure 3.4. A more sensible model using $\beta_1=1$ and $\beta_2=\beta_1/\sqrt{2}$ gives Figure 3.5 which appears satisfactory. Figure 3.6 is obtained using ICM with $\beta_1=4$ and $\beta_2=\beta_1/\sqrt{2}$; using such large values discourages sharp edges or protrusions in the image and this is illustrated clearly here.

Although Besag suggests the use of the closest mean classifier as the initial estimate ICM may be applied using any colouring as the initial estimate and the effect of this can be seen in Figures 3.7 and 3.8 which were obtained using ICM

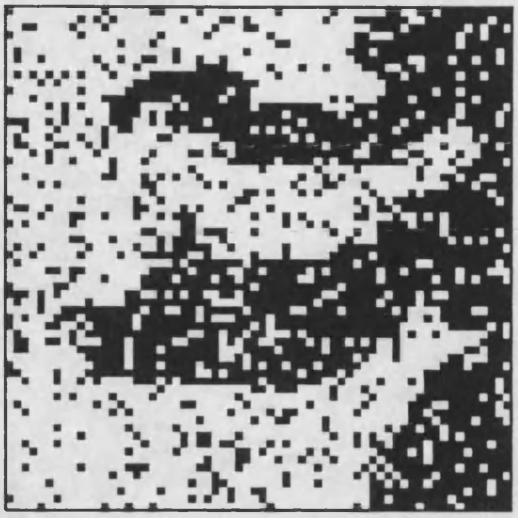


Figure 3.2
The closest mean classifier
Number of pixels misclassified : 673

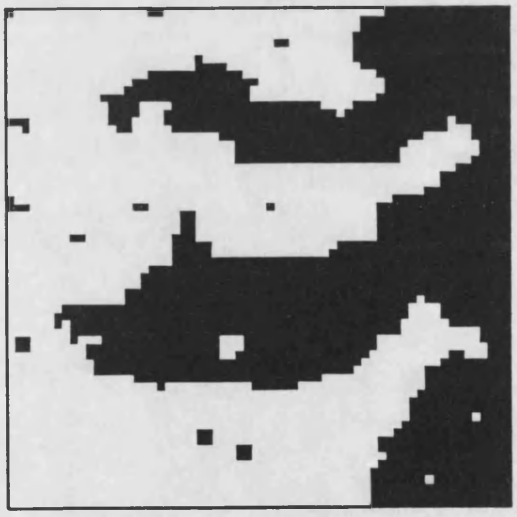


Figure 3.3
Reconstruction technique: ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = 0.0$ (first order model).
Initial estimate: Closest mean classifier
Number of pixels misclassified : 93

Figure 3.4
Reconstruction technique: ICM
Parameters: $\beta_1 = 0.0$
$\beta_2 = 1.0$
Initial estimate: Closest mean classifier
Number of pixels misclassified : 164

Figure 3.5
Reconstruction technique: ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \frac{\beta_1}{\sqrt{2}}$
Initial estimate: Closest mean classifier
Number of pixels misclassified : 62

Figure 3.6
Reconstruction technique: ICM
Parameters: $\beta_1 = 4.0$
$\beta_2 = \frac{\beta_1}{\sqrt{2}}$
Initial estimate: Closest mean classifier
Number of pixels misclassified : 142

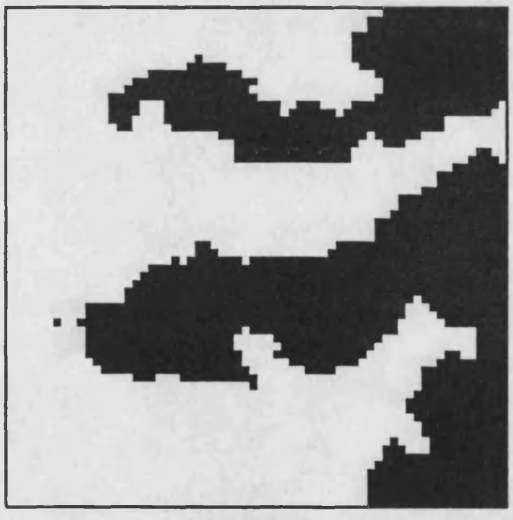


Figure 3.7
Reconstruction technique: ICM
Parameters: $\beta_1 = 0.8$
$\beta_2 = \frac{\beta_1}{\sqrt{2}}$
Initial estimate: All white scene
Number of pixels misclassified : 178

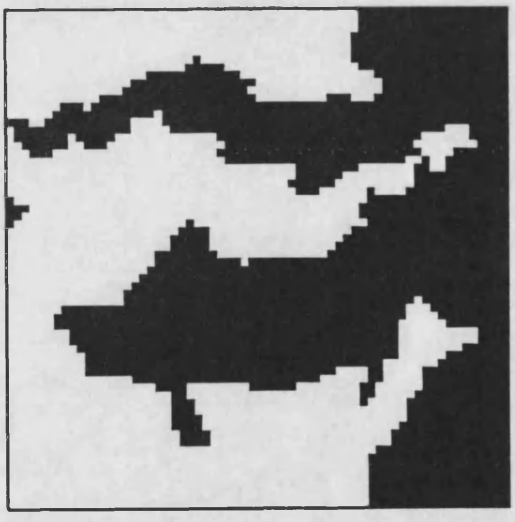


Figure 3.8
Reconstruction technique: ICM
Parameters: $\beta_1 = 0.8$
$\beta_2 = \frac{\beta_1}{\sqrt{2}}$
Initial estimate: All black scene
Number of pixels misclassified : 244

with $\beta_1=0.8$, $\beta_2=\beta_1/\sqrt{2}$ and initial estimates which were all white and all black respectively. When larger values of β_1 and β_2 were used the reconstruction varied little from the initial estimates in both cases.

When processing large images we avoid unnecessary computation by storing the coordinates of pixels whose colourings have changed in the current iteration. If the number of these is small, only pixels whose neighbours have changed colour in the last iteration are considered in the next iteration. In general, only two complete iterations are required in reconstruction although Figures 3.7 and 3.8 required 6 complete iterations before the number of changes became sufficiently small.

3.3 Simulated annealing

Geman and Geman (1984) liken images to physical systems in which the states of atoms or molecules interact in a lattice-like structure. In the physical context of annealing an energy is minimised by gradually reducing the temperature in the system; a gradual cooling is essential to avoid ending in a local minimum. The objective function is viewed in a similar way and random changes are allowed in the reconstruction process which increase the objective function in the hope that a better minimum will eventually be achieved. The essence of their approach uses a "stochastic relaxation" algorithm which generates a sequence of images that converges to the MAP estimate with probability 1.

The image is scanned in what is typically a raster fashion although the pixels may be considered for updating in any order as long as all pixels are visited infinitely often. Each n visits is known as a sweep of the image. Considering one pixel at any given time, the posterior probability $p_i(l)$ of colouring l at pixel i given the current colouring of the rest of the region is calculated for each of the $l=1,\ldots,L$ different colourings. Simulated annealing requires that colouring k is then assigned to pixel i with probability

$$\frac{\{p_i(k)\}^{\frac{1}{T}}}{\sum_{l=1}^{L}\{p_i(l)\}^{\frac{1}{T}}}$$

where T is the "temperature" of the process. The use of a temperature and cooling schedule follows directly from the physical analogy where gradual cooling isolates

low energy states which correspond to the most probable states in the posterior distribution. Theoretically the algorithm must run for an infinite amount of time with a logarithmic decrease in T for convergence to the global maximum to be guaranteed (see Gidas, 1985a). In practice a sensible cooling schedule must be used and truncated at some point in order that a reconstruction may be obtained. Geman and Geman advocate the use of the schedule

$$T(k) = \frac{C}{\log(1+k)} \qquad 1 \leq k \leq K$$

where K is the total number of sweeps and C is a fixed constant, usually taken to be between 2 and 5. Geman and Geman recommend the use of $C=3$ or 4.

The example is used to illustrate the effect of these parameters and the use of the closest mean classifier as an initial estimate of the true scene. Figure 3.9 shows the reconstruction obtained using the objective function (3.1) with $\beta_1=1.0$, $\beta_2=\beta_1/\sqrt{2}$, $C=3.5$, $K=1000$ and the closest mean classifier as an initial estimate of the true scene. These are typical values which we have found to give good results in most cases.

We first examine the effect of the C parameter. If the temperature is large then changes in pixel colourings depend less on the posterior probabilities and are more random. If the temperature is small then the changes are biased towards colourings which reduce the value of the objective function. When C is large the temperature is very high at the start of the process, increasing the likelihood that groups of pixels will change colour, leading to a different minimum of the objective function. If in the final sweep, the temperature remains comparatively high, changes may occur which significantly increase the value of the objective function, producing an image which has lower probability. This is illustrated in Figure 3.10 where $C=7.0$; the isolated pixel colourings occur as a result of changes made in the final sweep.

The value of K, the total number of sweeps, is also of interest. The theory which is used to justify the use of annealing suggests that the process should be allowed to run for as long as possible to maximise the chances of reaching the MAP estimate. In practical applications, where the size of the region and the number of different colourings of each pixel increase the computational burden, truncation must occur at some point. Figures 3.11 and 3.12 show the images obtained using $K=33$ and 10000 respectively. (We use the value $K=33$ in order

Figure 3.9
Reconstruction technique: Simulated Annealing
Parameters: $\beta_1 = 1.0$
$\beta_2 = \frac{\beta_1}{\sqrt{2}}$
$C = 3.5$
$K = 1000$
Initial estimate: Closest mean classifier
Number of pixels misclassified : 57

Figure 3.10
Reconstruction technique: Simulated Annealing
Parameters: $\beta_1 = 1.0$
$\beta_2 = \frac{\beta_1}{\sqrt{2}}$
$C = 7.0$
$K = 1000$
Initial estimate: Closest mean classifier
Number of pixels misclassified : 88

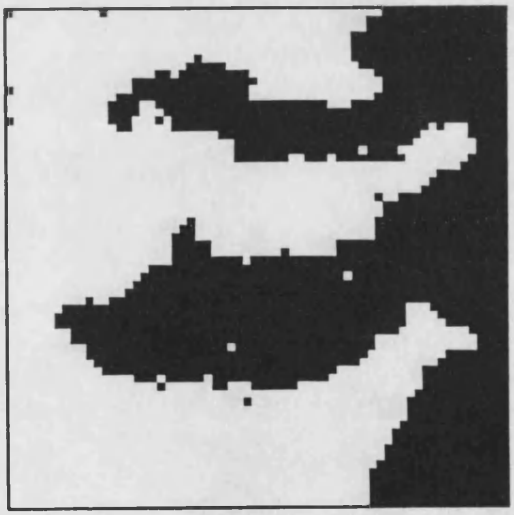


Figure 3.11
Reconstruction technique: Simulated Annealing
Parameters: $\beta_1 = 1.0$
$\beta_2 = \frac{\beta_1}{\sqrt{2}}$
$C = 3.5$
$K = 33$
Initial estimate: Closest mean classifier
Number of pixels misclassified : 68

Figure 3.12
Reconstruction technique: Simulated Annealing
Parameters: $\beta_1 = 1.0$
$\beta_2 = \frac{\beta_1}{\sqrt{2}}$
$C = 3.5$
$K = 10000$
Initial estimate: Closest mean classifier
Number of pixels misclassified : 56

that the temperatures at the final sweep for this reconstruction and the reconstruction shown in Figure 3.10 were approximately equal.) In general we have found that better reconstructions are obtained when K is chosen as large as possible.

We have already demonstrated the dependence of ICM on the initial estimate. Figures 3.13 and 3.14 show the reconstructions obtained by simulated annealing using all white and all black initial estimates respectively. Although these scenes differ slightly the initial estimate does not appear to have had a detrimental effect in either case. Clearly, if the process was terminated earlier any effect would be more obvious.

Isolated pixel colourings like those shown in Figures 3.10 and 3.11 may be avoided if, after the K sweeps, an additional sweep is executed with $T=0$. This is equivalent to one iteration of Iterated Conditional Modes and can only reduce the value of the objective function. Figures 3.15 and 3.16 show the reconstructions when this extra sweep has been applied to Figures 3.9 and 3.11 respectively.

3.4 Exact MAP estimation for binary images

Grieg, Porteous and Seheult (1989) show that for a two colour scene the MAP estimate may be found exactly. The problem is reformulated as a minimum cut problem in a capacitated network and the Ford-Fulkerson labelling algorithm is used to find the solution.

The likelihood function $l(y\,|\,x)$ is rewritten as

$$\prod_{i=1}^{n} f(y_i\,|\,x_i) = \prod_{i=1}^{n} f(y_i\,|\,x_i=1)^{x_i} f(y_i\,|\,x_i=0)^{1-x_i}$$

and the prior model as

$$p(x) \propto \exp\left[\frac{1}{2}\sum_{i=1}^{n}\sum_{j\in\partial_i}\beta_{ij}\mathrm{I}(x_i=x_j)\right]$$

where $\beta_{ij}=\beta_{ji}$ and $\mathrm{I}(x_i=x_j)=1$ if $x_i=x_j$ and 0 otherwise. Ignoring an additive constant we may write

$$\ln P(x\,|\,y) = L(x\,|\,y) = \sum_{i=1}^{n} x_i\lambda_i + \frac{1}{2}\sum_{i=1}^{n}\sum_{j\in\partial_i}\beta_{ij}\mathrm{I}(x_i=x_j)$$

where $\lambda_i = \ln\{f(y_i\,|\,x_i=1)/f(y_i\,|\,x_i=0)\}$, the log-likelihood ratio at pixel i.

Figure 3.13
Reconstruction technique: Simulated Annealing
Parameters: $\beta_1 = 1.0$
$\beta_2 = \frac{\beta_1}{\sqrt{2}}$
$C = 3.5$
$K = 1000$
Initial estimate: All white scene
Number of pixels misclassified : 61

Figure 3.14
Reconstruction technique: Simulated Annealing
Parameters: $\beta_1 = 1.0$
$\beta_2 = \frac{\beta_1}{\sqrt{2}}$
$C = 3.5$
$K = 1000$
Initial estimate: All black scene
Number of pixels misclassified : 51

Figure 3.15
Reconstruction technique: Simulated Annealing including an additional sweep with $T=0$
Parameters: $\beta_1 = 1.0$
$\beta_2 = \frac{\beta_1}{\sqrt{2}}$
$C = 3.5$
$K = 1000$
Initial estimate: Closest mean classifier
Number of pixels misclassified : 51

Figure 3.16
Reconstruction technique: Simulated Annealing including an additional sweep with $T=0$
Parameters: $\beta_1 = 1.0$
$\beta_2 = \frac{\beta_1}{\sqrt{2}}$
$C = 3.5$
$K = 33$
Initial estimate: Closest mean classifier
Number of pixels misclassified : 51

Consider a capacitated network containing a source, a sink and n interior vertices called nodes. There are a number of arcs from the source to interior nodes which allow flow in that direction. A number of interior nodes are connected to the sink by arcs which allow flow in that direction. In addition to these arcs, flow is permitted along arcs between interior nodes. Each of the arcs in the network has a capacity. This is the maximum flow in either direction along the arc. An example of a network is shown in Figure 3.17 although the capacities of the arcs are not shown.

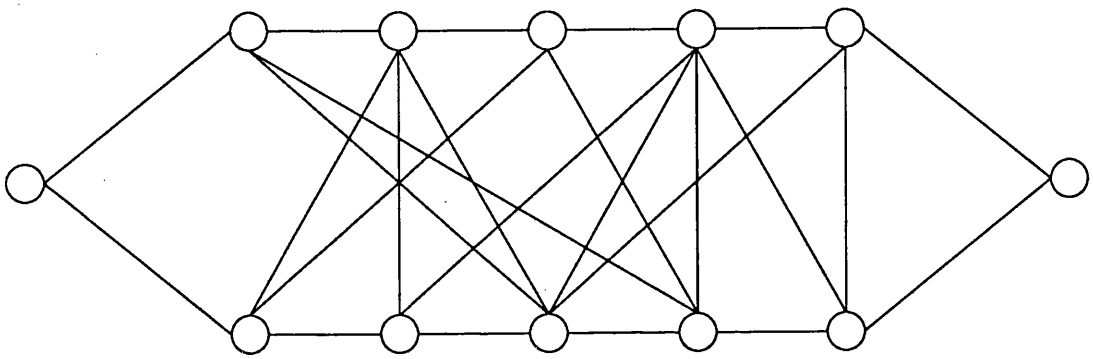


Figure 3.17

We define a *cut* of the network to be a partition of all the nodes into two sets. The first of these sets, S_1, contains the source and the other, S_2, contains the sink. Every interior node is contained in just one of the sets. The value of the cut is given by

$$C(S_1,S_2)=\sum_{k\in S_1}\sum_{l\in S_2} c_{kl}$$

where c_{kl} is the capacity of the arc between node k and node l.

The maximum flow problem is to maximise the flow from the source to the sink, using the arcs in the network. Ford and Fulkerson (1962) show that the maximum flow is equal to the minimum cut and provide an algorithm for finding the maximum flow.

In the context of imaging, the problem may be reformulated in the following way. The n pixels are regarded as interior nodes. Node i is connected to the source by an arc with capacity λ_i if $\lambda_i>0$, otherwise it is connected to the sink by an arc with capacity $-\lambda_i$. Node i is connected to node j by an arc with capacity β_{ij}. We refer to the source and sink as nodes s and t respectively. For any binary image x we may form the sets $B=\{s\}\cup\{i;x_i=1\}$ and $W=\{t\}\cup\{i;x_i=0\}$. These

two sets define a partition of the network which is a cut. The value of the cut is

$$C(x)=C(B,W)=\sum_{k\in B}\sum_{l\in W}c_{kl}$$

which may be written as

$$\sum_{i=1}^{n}x_i\ \max(0,-\lambda_i)+\sum_{i=1}^{n}(1-x_i)\ \max(0,\lambda_i)+\sum_{i=1}^{n}\sum_{j\in\partial_i}\beta_{ij}\mathrm{I}(x_i\neq x_j)$$

which differs from $-L(x|y)$ by a term which does not depend on x. The image x which maximises $L(x|y)$ corresponds exactly to the partition which gives the minimum cut, the set S_1 containing nodes corresponding to black pixels and S_2 containing nodes corresponding to white pixels. Hence, the MAP image estimate may be found by maximising the flow in the related network problem.

The Ford Fulkerson labelling algorithm for maximising the flow through a network

We use f_{ij} to represent the flow from node i to node j and denote the *excess capacity* of this arc by $d_{ij}=c_{ij}-f_{ij}$.

Step 0. Set $f_{ij}=0\ \forall ij$. Define the index of the source to be 0. Label the source with the ordered pair $(-1,\infty)$.

Step 1. We form the set N_1 of nodes which are connected to the source by arcs with positive excess capacities. We use the index m for a general node in N_1. Label each of the nodes in N_1 with the ordered pair (e_m,p_m), where

$$e_m=d_{0m}$$
$$p_m=0$$

Step 2. We form the set N_2 of all unlabelled nodes which are connected to nodes in N_1 by arcs with positive capacities in the following way: Choose the node in N_1 with the smallest index; say it is node k. Add to N_2 the **unlabelled** nodes which are joined to node k by arcs with positive excess capacities. We use the index m for a general node in N_2. Label node m in N_2 with the ordered pair (e_m,p_m), where

$$e_m=\min\{d_{km},e_k\}$$
$$p_m=k$$

Observe that e_m is the minimum of the excess capacities of the arcs from the source to node k and from node k to node m. Also, p_m denotes the node which led to node m. Repeat this for all nodes in N_1.

Step 3. Repeat Step 2 with N_r replacing N_2 and N_{r-1} replacing N_1, for $r=3,4,\ldots$ After a finite number of steps, we arrive at one of two possibilities:
(i) The sink has not been labelled and no other nodes can be labelled.
(ii) The sink has been labelled.

Step 4. If we are in case (i), then the current flow can be shown to be optimal and we stop.

Step 5. If we are in case (ii), we may now increase the flow. If the sink has the label (e_r,p_r) then the flow may be increased by e_r; the second label p_r indicates the node which led to the sink. Thus the route may be retraced back to the source and the flow increased by e_r in every arc along the route from source to sink.

Step 6. Remove all of the labels from the nodes and return to Step 1.

3.4.1 Improvements in the algorithm when used for MAP estimation.

The Ford Fulkerson labelling algorithm may be applied to general networks. In the context of imaging, the networks that are derived have a particular structure and it is this that may be exploited to reduce the amount of CPU time required to find the minimum cut. Figure 3.18 shows another network which may be compared with the network in Figure 3.17. The differences in the structure of these networks is clear. The network in Figure 3.18 is of the type that will be encountered in imaging problems. All internal nodes are connected to either the source or the sink and typically approximately half will be connected to each.

The main disadvantage of the algorithm is that once a path from the source to the sink has been found and the capacities updated, other incomplete paths will be discarded as the process begins again. Figure 3.19 shows the paths that have been found up to Step 3 of the algorithm, unused arcs are shown as dotted lines. The information supplied by all but one of these will be discarded. When the number of pixels in the region is large Step 1 of the algorithm must be repeated many times and the bulk of the information which is found is not used. Figure

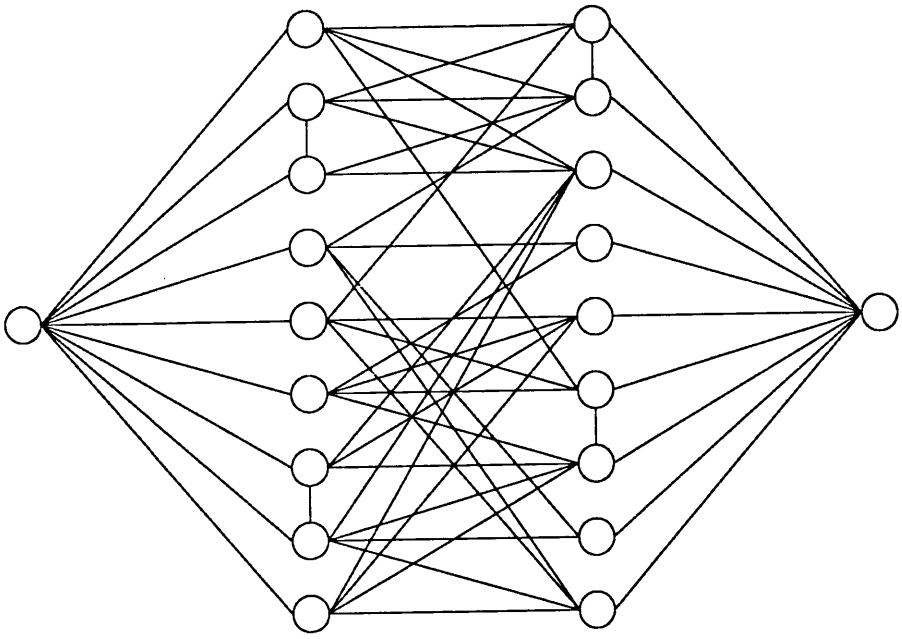


Figure 3.18

3.19 illustrates that there were several paths which would have reached the sink had the process been continued. Notice also that most of these paths would have been unaffected by updating the capacities of the successful path.

Grieg, Porteous and Seheult achieved a twelve fold reduction in CPU time by using the algorithm to find the MAP estimate for sub-regions of the image; they combine these sub-regions and then find the MAP estimate for the whole scene. In terms of the network, sections are considered in isolation and the flow maximised in that section. (Each section includes the source and the sink.) When the flow in each section is maximal all previously restricted arcs are opened and the flow increased in the original network. Figure 3.20 shows a way in which the network in Figure 3.18 may be divided up. Note that some of the arcs between internal nodes have been omitted. It is easy to see why this modification reduces the CPU requirement: if the region is small then Step 1 of the algorithm is shortened significantly.

We now propose a modified version of the labelling algorithm which exploits the special structure of the networks found in the image analysis context.

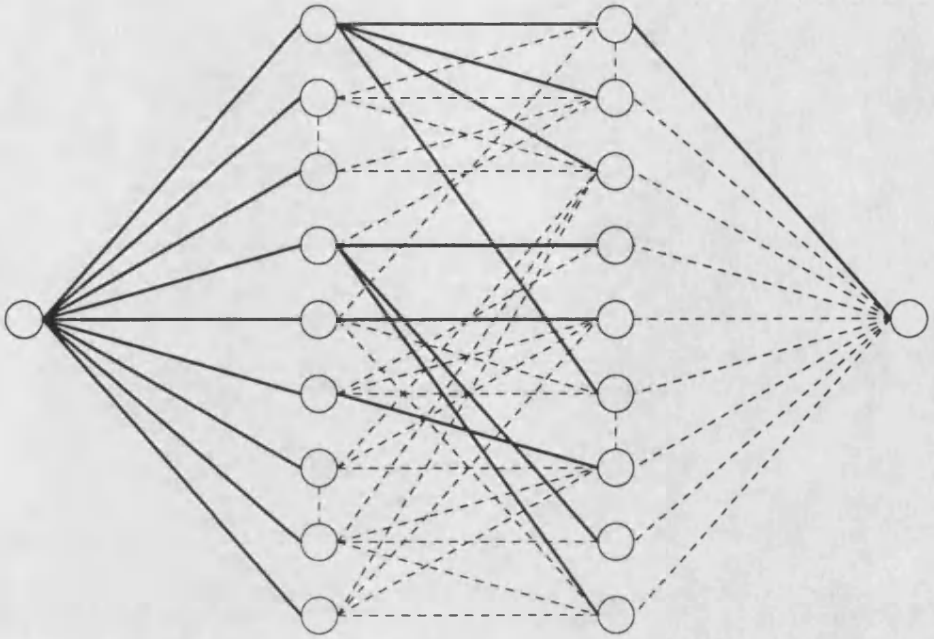


Figure 3.19

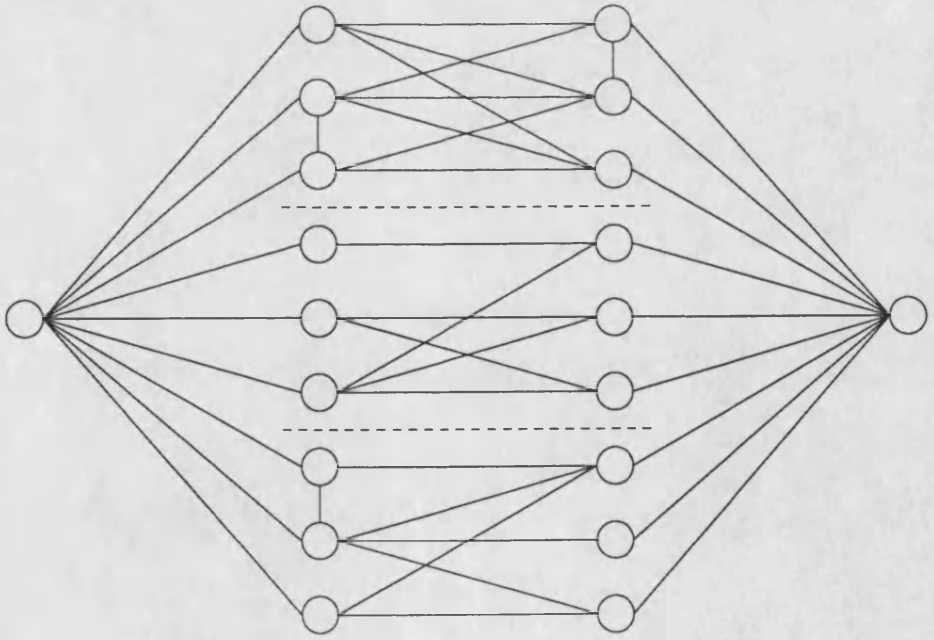


Figure 3.20

The Modified Algorithm

Step 0 Set $f_{ij}=0\ \forall ij$. Define the index of the source to be 0. Label the source with the ordered pair $(-1,\infty)$.

Step 1 We form the set N_1 of nodes which are connected to the source by arcs with positive excess capacities. Label each of the nodes in N_1 with 0.

Step 2 We form the set of all unlabelled nodes which are connected to nodes in N_1 by arcs with positive capacities the following way:

Choose the node in N_1 with the smallest index; say it is node k. Add to N_2 the unlabelled nodes which are joined to node k by arcs with positive excess capacities. If N_2 contains the sink then remove the sink from N_2 and add k to the set T of nodes which are the penultimate nodes on successful paths from the source to the sink. Consider all nodes in N_1 in this way. Label each unlabelled node in N_2 with k, the node which led to that node.

Step 3 Repeat Step 2 with N_r replacing N_2 and N_{r-1} replacing N_1, for $r=3,4,\ldots$ After a finite number of steps no other nodes can be labelled.

Step 4 If the set T contains no nodes then the current flow can be shown to be optimal and we stop. Otherwise we increase the flow in the following way:

For the first node in T we retrace the path that led to this node and and find the minimum of the excess capacities of the arcs on this path and the capacity of the arc from this node to the sink. The excess capacities on this path may now be updated. Consider now the second node in T, we again retrace the path and find the minimum excess capacity of the path from the source to the sink for which this is the penultimate node. Note that this minimum excess capacity may have been reduced in updating the path to the first node in T. If the minimum capacity is greater than zero then we update the excess capacities on this path. All the remaining nodes in T are also considered in this way.

Step 5 Return to Step 1.

The reduction in CPU time that will be achieved using this algorithm will depend chiefly on the number of paths that are found from the source to the sink whose minimum flow is unaffected by increasing the flow in other paths.

Suppose that two complete paths share a common arc and that this arc has the minimum capacity for one or both of the paths. When the flow is updated in the first path the minimum capacity of the second path has been reduced and thus the flow may not be increased by as much. We use the network in Figure 3.18 to illustrate this difficulty. Figure 3.21 shows the paths that have been found to Step 3 of the algorithm, unused paths are shown as dotted lines. If the capacities of all the arcs are equal then of the nine paths that we have found we can increase the flow in only 4 of these. We now propose a further modification to Step 2 of the algorithm.

Step 2 Let N_2 denote the set of all unlabelled nodes which are joined to nodes in N_1 by arcs with positive excess capacities. N_2 is formed in the following way:

Choose the node in N_1 with the smallest index; say it is node k. Add to N_2 *only one* unlabelled node which is joined to node k by an arc with positive excess capacity. If N_2 contains the sink then remove it from N_2 and add k to the set T of nodes which are the penultimate nodes on successful paths from the source to the sink. Consider all nodes in N_1 in this way. When all the nodes in N_1 have been considered for one connection to an unlabelled node we return to the beginning of the set and consider all nodes for a further connection. We repeat this until no more connections are possible. We use the index m for each node in N_1. Label each unlabelled node in N_1 with k, the node which led to node m.
Observe that this modification will give identical sets $N_1,\ldots,N_r$ to those found using the previous algorithm. It will, however, lead to different choices of paths between these nodes. Figure 3.22 shows the paths that may be found if this new method is adopted. In the example that we have shown, all of the complete paths are distinct and this increases the efficiency of the algorithm.

Results

Figure 3.23 shows the MAP estimate for the example discussed earlier in this chapter. It was obtained using $\beta_1=1.0$ and $\beta_2=\beta_1/\sqrt{2}$ and has only 49 misclassified pixels. The computer program that has been implemented is capable of using the raw Ford-Fulkerson algorithm, the partitioned version suggested by Grieg *et al.* and the modified algorithm both with and without partitioning. The estimate shown in Figure 3.23 can be obtained in each of these 4 ways and Table

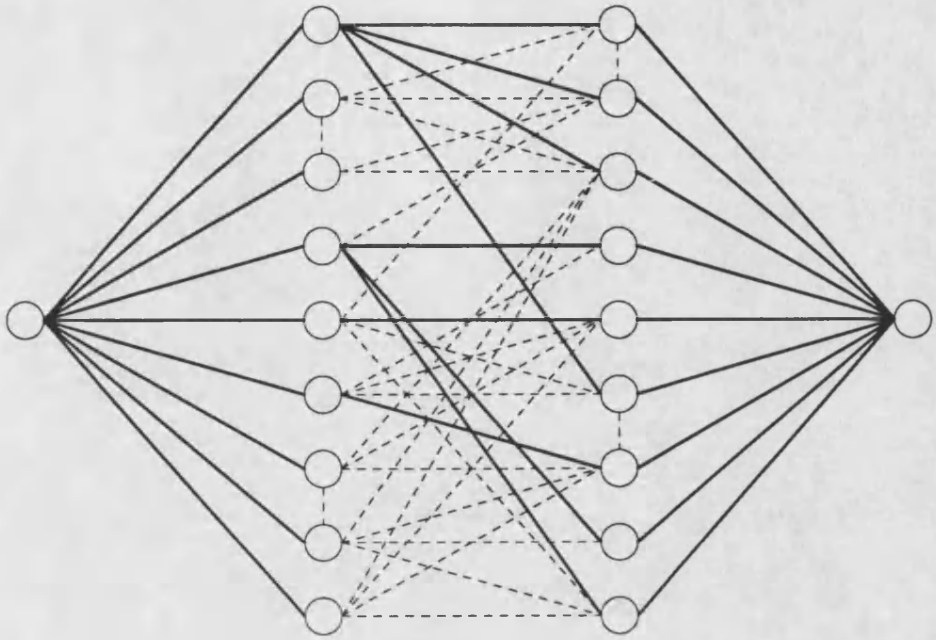


Figure 3.21

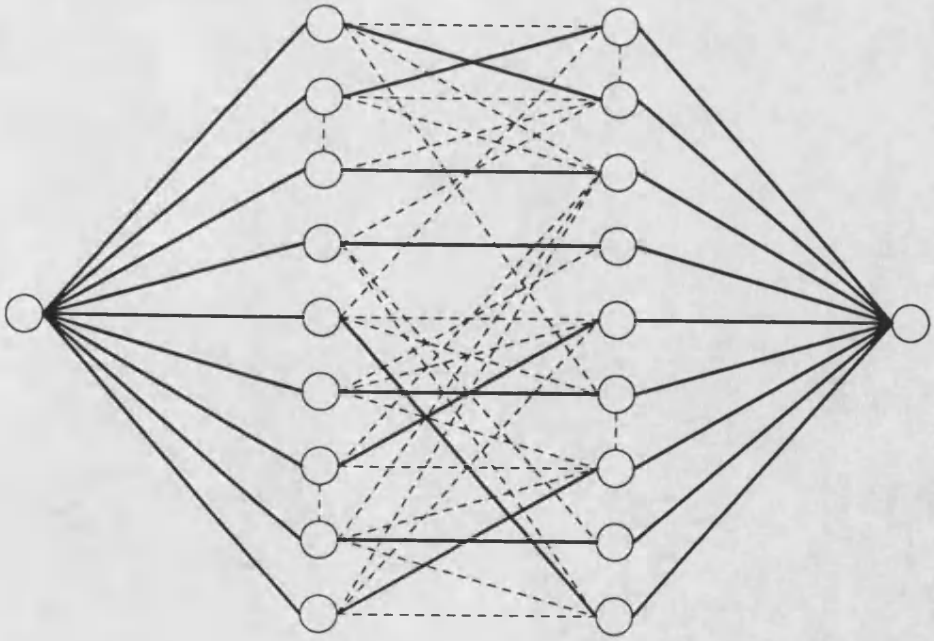


Figure 3.22

Figure 3.23

3.1 below shows the CPU requirement on a Sun 4 workstation for each case.

		using the partitioned version: Yes	using the partitioned version: No
using the modified algorithm	Yes	19.6	15.1
	No	73.0	770.0

Table 3.1

Seconds of CPU time required for a 64 by 64 binary image where $\sigma^2=0.25$, $\beta_1=1.0$ and $\beta_2=\beta_1/\sqrt{2}$.

In the partitioned version the flow is first maximised in separate 4 by 4 regions followed by 16 by 16 regions and then on the whole region. For the general m by n region we use approximately $m/16$ by $n/16$ regions followed by $m/4$ by $n/4$ regions and then the whole region, although if the sub regions are very small then we progress directly to the next level. Grieg *et al.* point out that any sensible choice will lead to a substantial reduction in CPU time. A larger example was considered to illustrate the massive reduction that may be obtained using the

modified algorithm. A 128 by 128 binary scene was perturbed by additive Gaussian noise with zero mean and variance 0.9105. This level of the variance gives a 30% misclassification rate in the closest mean classifier and was used in the examples shown in Grieg *et al.* (1989). The results are obtained using a second order neighbourhood with equal weightings for all of the neighbours and with β=0.3.

		using the partitioned version: Yes	using the partitioned version: No
using the modified algorithm	Yes	115	111
	No	630	15921

Table .2
Seconds of CPU time required for a 128 by 128 binary image where σ^2=0.9105 and $\beta_1=\beta_2$=0.3.

In both of the examples shown above the fastest results were obtained using the modified algorithm alone. Combining the partitioned version with the modified algorithm has little effect on the CPU requirement. The reduction obtained by Grieg *et al.* using the partitioned version can also be seen here. Both the variance and the value of β that is used affect the time required for processing. Where the variance is small the arcs from the source and sink to the internal nodes have smaller capacities and hence the flow may be maximised quicker. When β is small the capacities of the arcs between internal nodes are small and again the flow may be maximised quicker. For a 128 by 128 example with variance 0.1 and β=0.3 the modified algorithm takes only 49 seconds.

3.4.2 Comparing the approximate MAP reconstructions with the exact MAP estimate.

In Sections 3.2 and 3.3 we have compared the reconstructions that have been obtained with the original true scene. In addition we may now compare these reconstructions with the exact MAP estimate shown in Figure 3.23.

Figure	Number of different pixels
3.5	23
3.9	18
3.12	13
3.13	16
3.14	18
3.15	8
3.16	8

Table 3.3
Comparing earlier reconstructions with exact MAP estimate

These numbers allow us to assess directly the capability of different methods to find the MAP estimate, independently of whether the MAP estimate is itself a good estimate of the true image.

3.5 Other methods

The methods that have been described are not the only techniques for MAP estimation, although many recently developed techniques do incorporate ICM or Simulated Annealing. We now describe two MAP estimation techniques which use alternative approaches to the maximisation.

Derin *et al.* (1984) consider horizontal strips of the image and uses a dynamic programming algorithm to find the MAP estimate for each strip. These overlapping strips are combined to give a near optimal MAP estimate of the true scene. This technique depends heavily on the correlation between the colourings of pixels dropping rapidly as the vertical distance between them increases. If the strips are narrow then it is more likely that vertical correlations will exist and since the method can only feasibly be operated with a strip width of 2 to 4 pixels this may be a drawback. In subsequent work Derin *et al.* (1985) show that the

computational burden usually incurred when processing images containing several different region types may be avoided to some extent if the region types may be ordered. Consider a region which contains 4 region types with means r_1, r_2, r_3 and r_4; with $r_1<r_2<r_3<r_4$. Derin *et al.* first assume that the image contains only 2 region types and that these have means r_2 and r_3. The dynamic programming algorithm is applied and a reconstruction is obtained. The pixels which have been coloured with r_2 are then considered in isolation and the algorithm applied assigning colours r_1 and r_2. The region which was coloured with mean r_3 is then reconstructed using the colours r_3 and r_4. This gives a 4 colour reconstruction for the whole scene. Good results are obtained using this method. The examples that are shown in Derin *et al.* (1985) consider up to 8 levels but clearly the method may be extended to handle any number of levels. Combining this technique with exact MAP estimation (Greig *et al.*) as described in Section 3.4 we may obtain the exact MAP estimate at each two colour stage as compared with the near optimal estimate offered by Derin *et al.* (1985). Although the exact MAP estimate may be obtained at each stage, combining these estimates in this way will not give the global MAP estimate for the k level case. A method which uses this approach is described in Chapter 6.

Gidas (1989) proposed a "coarse-to-fine" method which reconstructs the image, using simulated annealing, on a coarser grid than that on which the record was collected. This reconstruction is then used in turn to supply information to the process at a finer level. This process is repeated until a reconstruction has been obtained at the scale at which the record was originally collected. Gidas states

> *"The intuitive basis of the method .... lies in the fact that in image processing problems, ... as well as in many other massive computational tasks we are facing today, one deals with cooperative features that occur over many spatial (or temporal) scales with various interscale interactions."*

and continues

> *"Furthermore, it is intuitively clear that multilevel multiresolution processing can reveal useful information for representation, interpretations, or recognition tasks."*

The method uses a renormalisation group algorithm which provides consistency between the model at different grid scales. We shall not describe the algorithm in more

detail here but in Chapter 4 of this thesis we propose a method which uses a similar "cascade" approach which incorporates ICM at each stage.

3.6 Parameter estimation

In the context of imaging there are in general two parameter vectors that may be unknown. The first of these is the noise parameter; where the distribution of the noise is known, say it is Gaussian, the mean and variance of the distribution must be estimated. For this case the mean is usually zero and where the variance is unknown it can often be accurately estimated from the study of training data.

In addition to any noise parameters that must be estimated we require a suitable value of β, the "smoothing" parameter which affects the smoothness of the reconstruction.

Coding methods

Cross and Jain (1983) generate Markov Random Fields using the Metropolis algorithm and have notable success in simulating binary and grey scale textures using parameters which have been estimated from observed textures. They use coding methods, first introduced by Besag (1972), to estimate the parameters. Given a colouring of the region, the pixels are divided into sets which do not contain neighbours, this division does not depend on the colourings of the pixels. For each set, estimates are obtained which maximise the product of the conditional likelihoods of the colourings of the pixels given the colourings of their neighbours. The advantage of this method is that under the assumptions of the model, the colourings of each of the pixels in any set are conditionally independent of one another given the colourings of all other pixels, and thus maximum likelihood estimates can be obtained from the conditional likelihoods. For a first order model the pixels are coded 1 or 2 such that pixels which are horizontally or vertically adjacent are coded differently, giving a chequerboard effect. Each coded region is then treated separately giving two sets of observations, each providing one estimate of the parameter. The estimates from the two sets may then be combined in an appropriate way to give one estimate. Higher order models require different codings and the estimates for different codings are obtained in the same way.

Besag (1986) suggests that a neater and more efficient procedure is to use maximum pseudo likelihood, which finds the estimate which maximises the product of the conditional likelihoods over all the pixels in the region. When maximum pseudo likelihood is used, Besag suggests that boundary pixels are excluded from the likelihood because of the artificiality of the model there. A more detailed overview of maximum pseudo likelihood is given by Geman and Graffigne (1986).

Pseudo-likelihood and maximum likelihood methods may be easily used in iterative reconstruction techniques. Such techniques are sometimes known as adaptive methods and are used in conjunction with simulated annealing by Lakshmanan and Derin (1989) and Geman (1985). Besag (1986) outlines a procedure for estimating both the noise parameters and a suitable smoothing parameter during ICM. Frigessi and Piccioni (1988) consider the case of the binary channel, where each pixel changes colour with unknown probability ε, independently of the others. They propose a method for finding estimates of both ε and β which they show are consistent if the region is regarded as having a free boundary. The consistency of maximum likelihood and pseudo likelihood estimates for MRF parameters is also examined by Gidas (1986).

As we have mentioned earlier a "try it and see" approach is commonly used and, where the method is computationally inexpensive, it may be desirable to produce several reconstructions using different β parameters.

It is important to understand the influence of the method which is being used to provide the reconstruction when considering values of β. Often very different reconstructions can be obtained from different methods using the same parameters. With a local technique like ICM, the reconstruction process is such that pixels which are far apart can rarely influence one anothers colourings. Where exact MAP estimation is used long range dependencies in the model can have a large affect on the resulting reconstruction.

In addition to the works that have already been mentioned the following references are considered useful texts in this area. For general overviews of the statistical problem see Grenander (1983), Ripley (1986) and Ripley (1988a). On the use of MRF models as priors see Besag (1974), Kashyap and Chellapa (1983) and Ripley (1988b). For an introduction to parameter estimation see Grenander (1985), Grenander and Osborn (1983) and Gidas (1985b). See also Hall and

Titterington (1986) on more general parameter estimation.

Chapter 4: Aggregation and the cascade algorithm

4.1 Aggregation

It is common for the record to be collected on very fine pixel grids; digitised images often measure 256 by 256 pixels or even larger. The size of such images allows the use of aggregation, a technique whereby sets of 2 by 2 pixels are replaced by a single large pixel with record equal to the average of the original four. This also corresponds to viewing the original image on a coarser grid. While the variance of the new record is one quarter that of the original, the range of the pixel colourings remains the same; thus the signal to noise ratio is substantially increased. The aggregated record may then be treated as the original and reconstructions obtained in the usual ways. The effect of reconstructing the image from an aggregated record rather than the original record depends on the reconstruction technique that is used. Because ICM is a local updating procedure pixel colourings are often only influenced by their close range neighbours. If we use the same MRF model on the aggregated record, pixels on the original grid (which now form part of the larger pixels) are more likely to be affected by pixels some distance away. Where simulated annealing or exact MAP estimation is used, pixels which are further apart can still have a substantial influence on each others' colourings. For this reason the effect of aggregation is not as apparent when these techniques are used. Whichever technique is applied, it can be seen that aggregation is a form of smoothing in itself, since it forces 2 by 2 squares of pixels to be the same colour. In this chapter we shall be concentrating on the use of aggregation when applied in conjunction with ICM, a technique which we have already shown to be fast and effective in handling moderate noise levels. We demonstrate that ICM can encounter severe problems when dealing with very noisy data and propose a technique which exploits the dependence of ICM on the initial estimate of the image.

We illustrate the use of aggregation with an example. Figure 4.1 shows a 256 by 256 binary scene. Gaussian noise with variance 4 is added to the data and the closest mean classifier is shown in Figure 4.2. Attempting to reconstruct the image at this grid level produces the very different reconstructions shown in Figures 4.3 and 4.4. Figure 4.3 was obtained using ICM: although the image appears blotchy, increasing the value of β_1 has little effect on the reconstructions

Figure 4.1
The true scene
Size 256 × 256

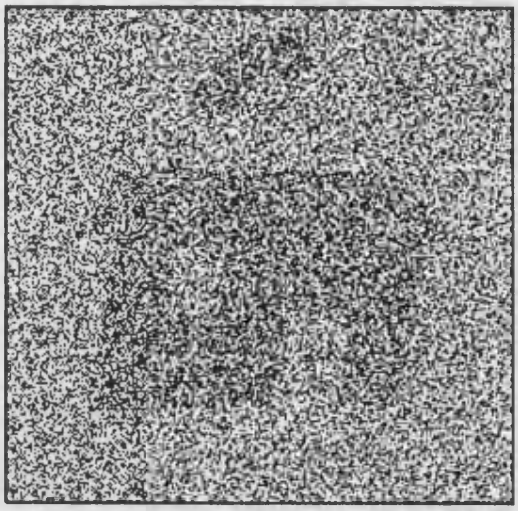


Figure 4.2
The closest mean classifier
Number of pixels misclassified : 26341

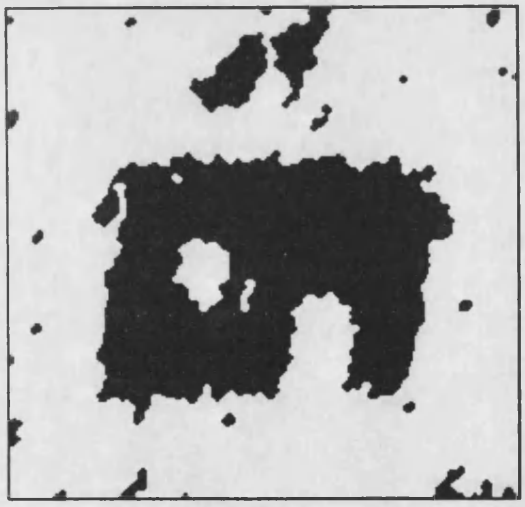


Figure 4.3
Reconstruction technique: ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: Closest mean classifier
Number of pixels misclassified : 4077

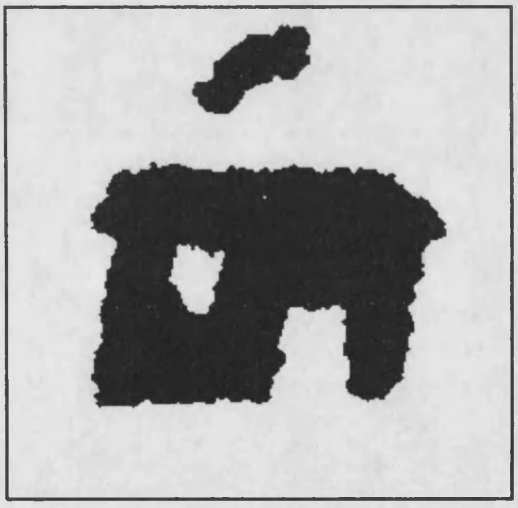


Figure 4.4
Reconstruction technique: exact MAP
Parameters: $\beta_1 = 0.3$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: Closest mean classifier
Number of pixels misclassified : 1348

obtained at this noise level. The exact MAP estimate is excellent, and has a misclassification rate of 2%. However, MAP estimates of such large regions are not easily obtained and we have added to the computational burden by searching for the value of β_1 which gave us a good estimate of the known true scene. The 256 by 256 record may be aggregated to give a 128 by 128 record with variance 1.0. Using the aggregated record we obtain the reconstructions shown in Figures 4.5 and 4.6. Repeating the aggregation process a 64 by 64 record with variance 0.25 is obtained. Figures 4.7 and 4.8 show the reconstructions for this record. Again the MAP reconstructions shown in Figures 4.4 and 4.6 have been obtained using a value of β_1 which gave a reconstruction which was close to the known true scene. In addition to the benefits of an increased signal to noise ratio there is also a significant reduction in the amount of processing required.

It is important to understand that the MRF models that are used at different levels of aggregation are not equivalent. There is a major change in the neighbourhood structures that are used at the two levels when considered in terms of the smaller pixel. Figure 4.9 shows the neighbours which are included in the MRF specification of the model at two levels of aggregation. For the finer grid, the shaded pixels are the neighbours of the black pixel. For the coarser grid, the black pixel must be coloured the same colour as the other subpixel which share the common larger pixel, and its colouring is affected by the colourings of the shaded pixels.

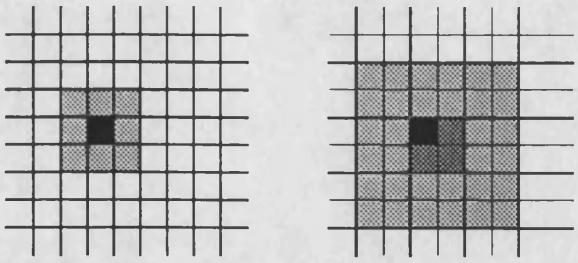


Figure 4.9

The two models are not equivalent, neither can they be made equivalent by a simple change in the parameters but a more detailed analysis is required. Gidas (1986) uses a single MRF model defined on the finest pixel grid and employs the "renormalisation group" approach to compute the models implied for the coarser grid. The models at the aggregated level are very complex and we would argue that, given the approximate nature of the MRF model, it is not unreasonable to

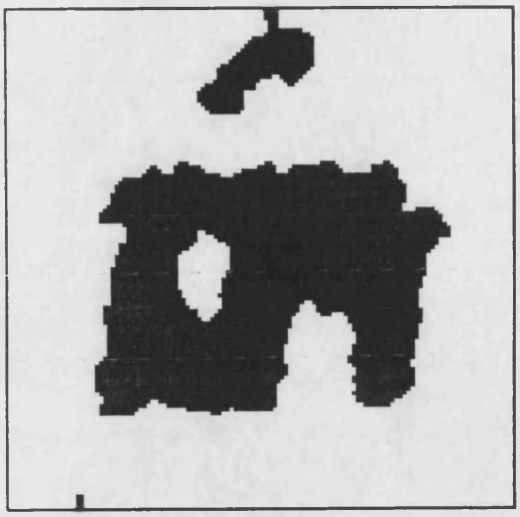


Figure 4.5
Reconstruction technique: ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: Closest mean classifier
Dimensions: 128 × 128
Record: 256 × 256 aggregated

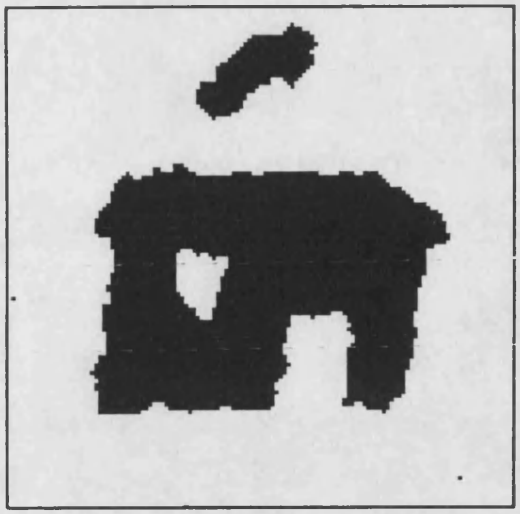


Figure 4.6
Reconstruction technique: exact MAP
Parameters: $\beta_1 = 0.5$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: Closest mean classifier
Dimensions: 128 × 128
Record: 256 × 256 aggregated

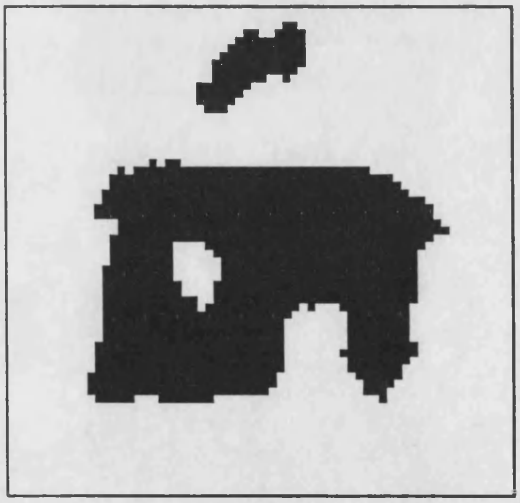


Figure 4.7
Reconstruction technique: ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: Closest mean classifier
Dimensions: 64 × 64
Record: 256 × 256 aggregated twice

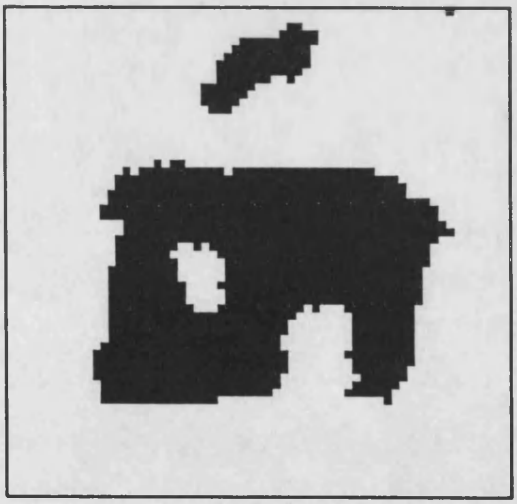


Figure 4.8
Reconstruction technique: exact MAP
Parameters: $\beta_1 = 0.8$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: Closest mean classifier
Dimensions: 64 × 64
Record: 256 × 256 aggregated twice

reduce computation at the expense of a less rigorous treatment of the model.

In some cases aggregation may not be helpful. For example, if the true scene contains regions of colour which are only a couple of pixels wide then aggregation may lead to their obliteration. However, scenes of this type are unlikely to give good reconstructions if the variance is high. The example illustrates that the reconstruction of large regions of colour may be greatly assisted using aggregation.

4.2 The cascade algorithm

In Section 3.2 we demonstrated the dependence of ICM on the initial estimate. In the above example we have seen that aggregation may be helpful in reconstructing the larger features of an image. We now propose a method which uses the reconstructions obtained on coarser grids as the initial estimates for ICM at finer levels.

A reconstruction is obtained by aggregating the record until it is one pixel in size. This is then used as the initial colouring for the ICM method on the 2 by 2 grid. This reconstruction is in turn used as the initial colouring for ICM on the 4 by 4 grid and we continue in this way, obtaining reconstructions right up to the level of the original record. We illustrate the success of this technique with two examples. The first of these is based on the scene shown in Figure 4.1 for which the closest mean classifier is shown in Figure 4.2. Recall that Gaussian noise with variance 4.0 has been added. The cascade reconstructions are shown in Figures 4.10 to 4.15. The true scene for the second example is shown in Figure 4.16. Gaussian noise with variance 8.0 is added to the data and the closest mean classifier is shown in Figure 4.17. The cascade reconstructions are shown in Figures 4.18 to 4.23. For both of these examples the reconstructions for the single pixel, the 2 by 2 grid and the 4 by 4 grid are not shown.

The algorithm works well in the first example. Figure 4.11 gives a good indication of the structure of the image and this structure is preserved through subsequent stages. We shall comment in Section 4.3 on the relatively small changes occurring in the final stages of the algorithm. The second example has been chosen to show the limitations of the method. The variance for this record is extremely high and some of the regions of colour are only a few pixels wide. The algorithm is successful in reconstructing the largest of the regions but fails to

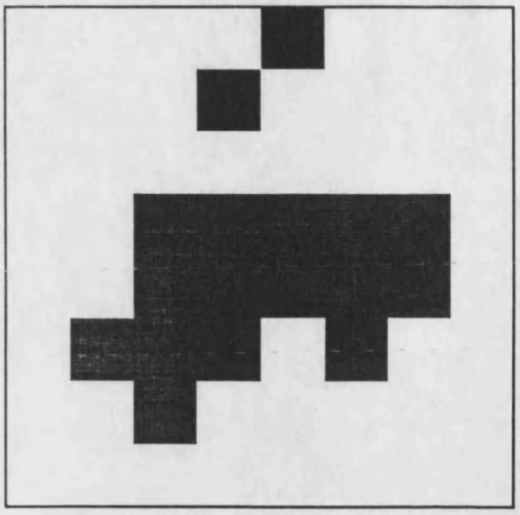


Figure 4.10
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 4 × 4 cascade
Dimensions: 8 × 8
Record: 256 × 256 aggregated 5 times

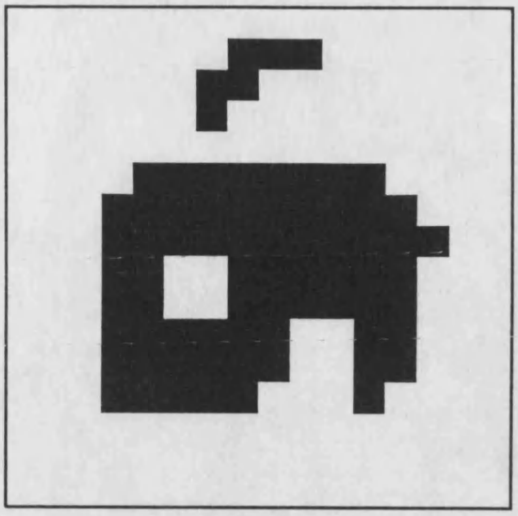


Figure 4.11
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 8 × 8 cascade (Fig 4.10)
Dimensions: 16 × 16
Record: 256 × 256 aggregated 4 times

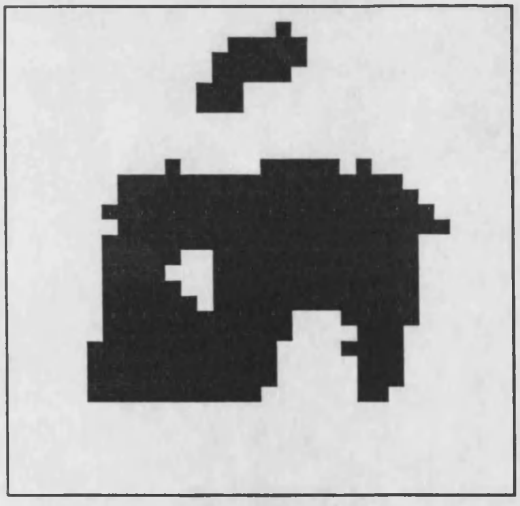


Figure 4.12
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 16 × 16 cascade (Fig 4.11)
Dimensions: 32 × 32
Record: 256 × 256 aggregated 3 times

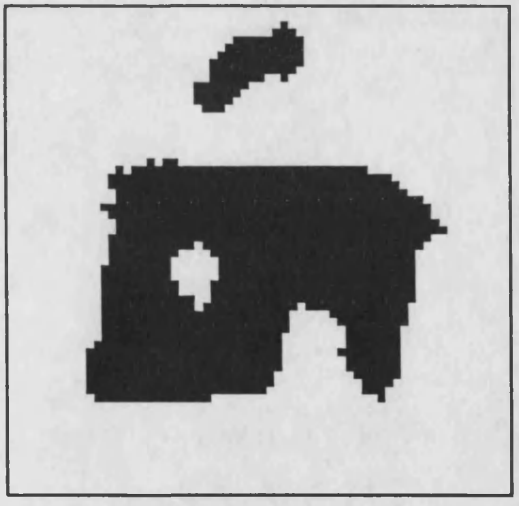


Figure 4.13
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 32 × 32 cascade (Fig 4.12)
Dimensions: 64 × 64
Record: 256 × 256 aggregated twice

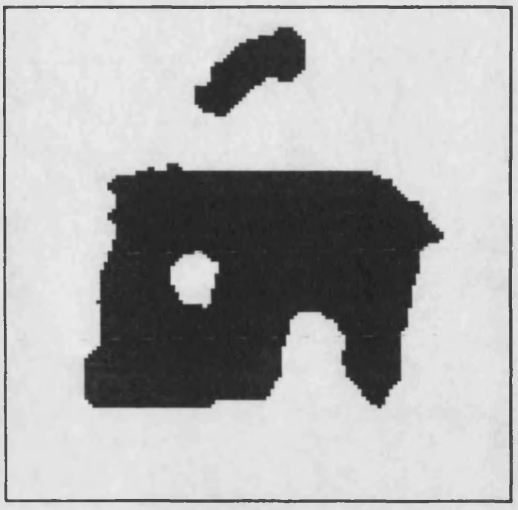


Figure 4.14
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 64 × 64 cascade (Fig 4.13)
Dimensions: 128 × 128
Record: 256 × 256 aggregated

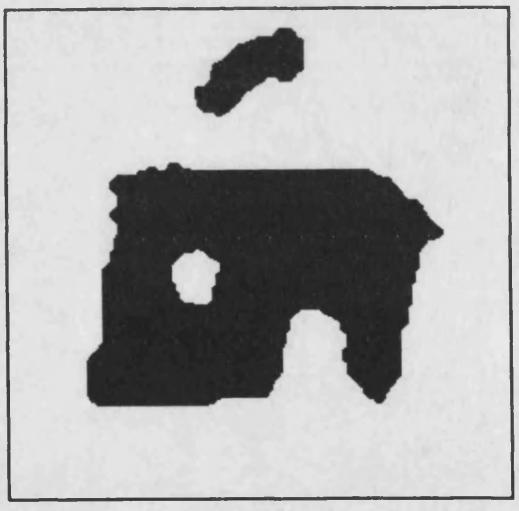


Figure 4.15
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 128 × 128 cascade (Fig 4.14)
Dimensions: 256 × 256
Record: 256 × 256 (original)
Number of pixels misclassified : 1928

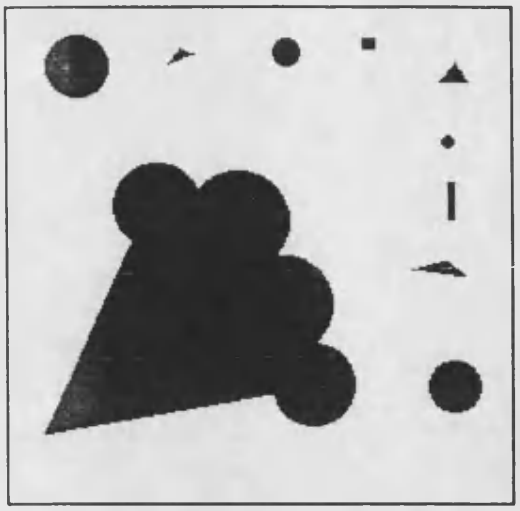


Figure 4.16

The true scene

Size 256 × 256

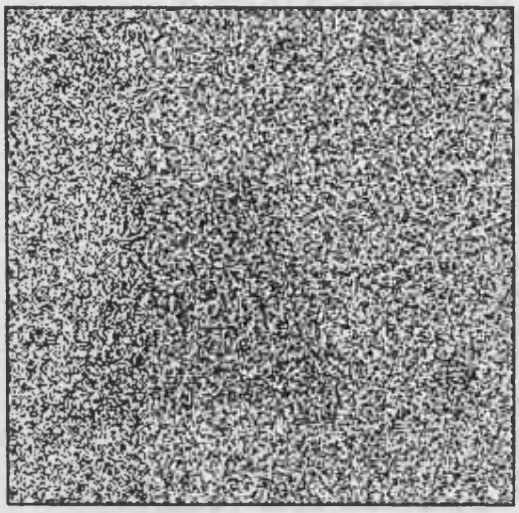


Figure 4.17

The closest mean classifier

Number of pixels misclassified : 28109

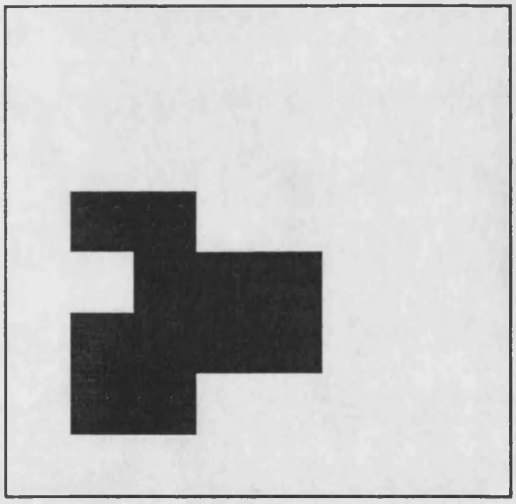


Figure 4.18

Reconstruction technique: Cascade, ICM

Parameters: $\beta_1 = 1.0$

$\beta_2 = \beta_1/\sqrt{2}$

Initial estimate: 4 × 4 cascade

Dimensions: 8 × 8

Record: 256 × 256 aggregated 5 times

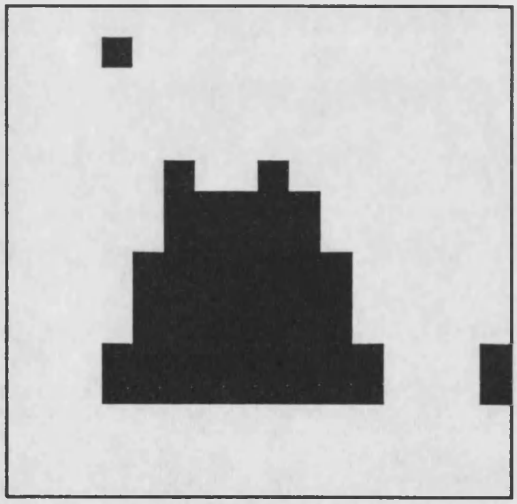


Figure 4.19

Reconstruction technique: Cascade, ICM

Parameters: $\beta_1 = 1.0$

$\beta_2 = \beta_1/\sqrt{2}$

Initial estimate: 8 × 8 cascade (Fig 4.18)

Dimensions: 16 × 16

Record: 256 × 256 aggregated 4 times

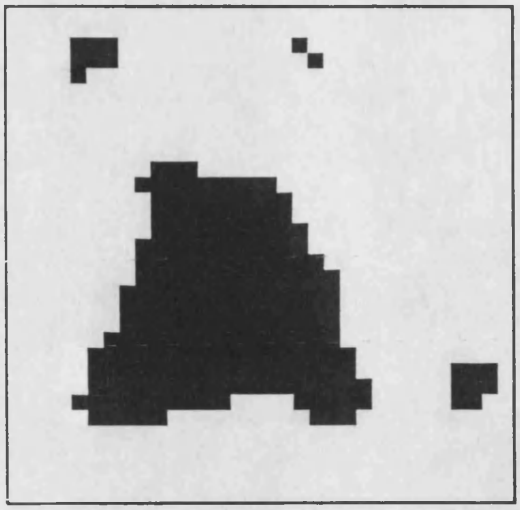


Figure 4.20
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 16 × 16 cascade (Fig 4.19)
Dimensions: 32 × 32
Record: 256 × 256 aggregated 3 times

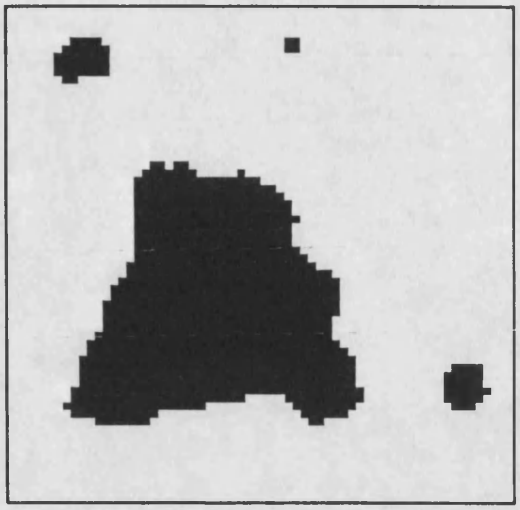


Figure 4.21
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 32 × 32 cascade (Fig 4.20)
Dimensions: 64 × 64
Record: 256 × 256 aggregated twice

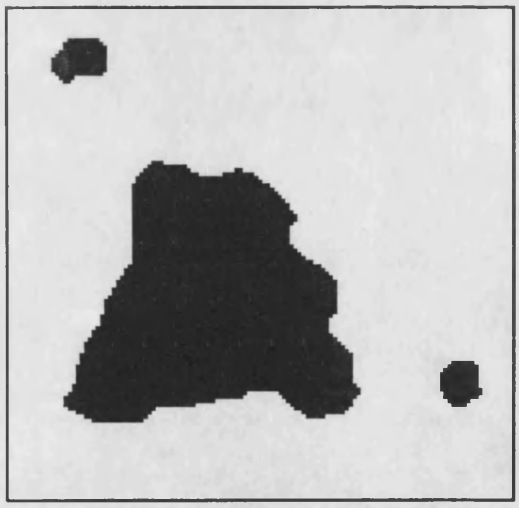


Figure 4.22
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 64 × 64 cascade (Fig 4.21)
Dimensions: 128 × 128
Record: 256 × 256 aggregated

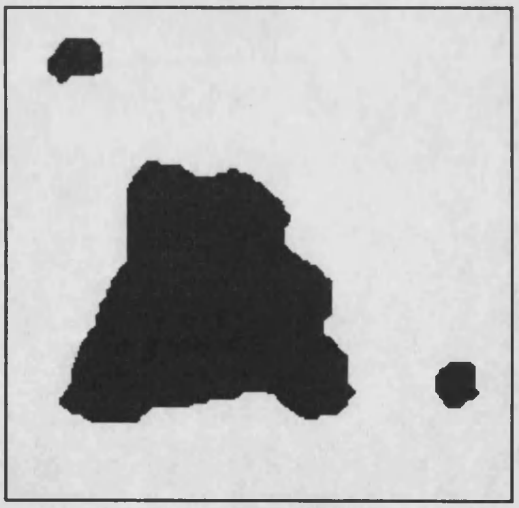


Figure 4.23
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 128 × 128 cascade (Fig 4.22)
Dimensions: 256 × 256
Record: 256 × 256 (original)
Number of pixels misclassified : 2075

reconstruct the smaller regions. It is interesting to note that one of the smaller regions appears in two of the intermediate stages but is later obliterated. This apparent phenomenon can be explained. In the early stages of the process the cascade reconstructions are very close to the closest mean classifier (in this example the variance for the 8 by 8 record is $8/4^5=0.0078125$) and so a region will not become apparent until it covers at least one half of a pixel. As the grid size increases so too does the variance and the trustworthyness of the individual records becomes more suspect. When the variance is large, as in the last two stages of the example, this leads to the obliteration of small regions of colour.

One might try to develop theoretical arguments to produce a "correct" sequence of values of β_1 for use at different stages of the cascade algorithm. In the context of edge processes, where the line edges which separate regions of colour are modelled as an alternative to discrepancies, Silverman, Jennison and Brown (1989) interpret the prior as a penalty and suggest that it should be chosen to be approximately independent of the pixel grid superimposed. They suggest that this penalty should approximate a constant multiple of the total boundary length in the image. In our application this would imply that the parameter β_1 be halved as the pixel sizes are quartered. However if we wish to preserve correlation between 2 points fixed in the image, then we would need to increase β_1 as the pixels get smaller. In Section 4.1 we found it best to use smaller values of β_1 at finer grid levels when exact MAP reconstruction was used. ICM behaves very differently and we have not found using smaller values at finer levels to be very successful. Although in some cases the range of "suitable" values of β_1 appears to change at different levels of aggregation, these ranges overlap and we have obtained good results using the same value of β_1 at all levels.

Again we can use the faster version of ICM to avoid unnecessary computation. Here we store the coordinates of pixels whose colourings have changed in the current iteration. If the number of these is small, only pixels whose neighbours have changed colour in the last iteration are considered for updating in the next iteration. For each of the cascade reconstructions one complete iteration plus some minor changes was all that was required. Summing a geometric series, we see that the total computation required is approximately equivalent to $1\frac{1}{3}$ complete iterations of ICM on the finest pixel grid.

In this section we have incorporated ICM into the cascade algorithm. The reasons for this were the strong dependence of ICM on the initial estimate and the intuitive appeal of using an aggregated reconstruction as the initial estimate. In addition, the computation required for processing the image is significantly reduced. It is possible that a different technique such as simulated annealing could be incorporated into the cascade algorithm but the initial estimate has little effect on the reconstruction obtained unless the number of sweeps is very small.

4.3 Early stopping of the cascade algorithm

Looking at the Figures 4.13, 4.14 and 4.15, we can see that although there are small changes in the details the overall structure of the image remains unchanged as finer levels are used. The Table 4.1 shows the number of changes in pixel colourings which occur at each stage for both of the examples. These figures are also shown as a percentage of the size of the region.

	1st example		2nd example	
Grid level	Number of changes	Percentage change	Number of changes	Percentage change
8 by 8	11	17.2	7	10.9
16 by 16	37	14.5	20	7.8
32 by 32	48	4.7	41	4.0
64 by 64	94	1.2	78	1.9
128 by 128	143	0.9	115	0.7
256 by 256	177	0.3	150	0.2

Table 4.1

These figures may be used as a subjective guide as to whether the continuation of the algorithm to next grid level is worthwhile given the additional computational burden. In both of the examples it is reasonable (with hindsight!) to consider stopping the process after the 128 by 128 reconstructions have been obtained.

4.4 Cascade and grey level reconstruction

We now consider the use of the cascade algorithm when the image contains more than 2 colours, now thought of as grey levels. We introduce a prior which may be used in conjunction with ICM to successfully reconstruct certain types of grey level images. We then incorporate this method into the cascade algorithm and obtain very good results.

4.4.1 A grey level problem and prior

Consider a scene in which each pixel takes one of R grey levels; typically R=64 or 256. The record y_i at each pixel is perturbed by additive independent Gaussian noise with mean 0 and variance σ^2. For problems of this type the choice of the prior can prove crucial. Different priors exhibit different qualities in reconstructions and the choice of the prior will depend on the expected qualities of the true scene. We assume that in the true scene there are k underlying levels $l_1, l_2, \ldots, l_k$ and that the region is separated into smaller regions, where each sub region takes one of the underlying levels with possibly some within level fluctuation, i.e. inside an area which takes one of the underlying grey levels, l_r say, there are pixels which are coloured with grey levels very close to l_r. We aim to reconstruct the regions and are not concerned with the preservation of any fluctuations. These assumptions will not apply to all applications; e.g. if we expect very smooth transitions between grey level regions. Because of the nature of the problem we choose to use the prior given by

$$p(x) \propto \exp\left\{-\frac{\beta}{2}\sum_{i=1}^{n}\sum_{j\in\partial_i}\left[1-\frac{1}{1+\alpha(x_j-x_i)^2}\right]\right\}. \qquad (4.4.1)$$

This prior was introduced in the context of single photon emission tomography (SPET) by Geman and McClure (1985). Inspection of the term

$$\left[1-\frac{1}{1+\alpha(x_j-x_i)^2}\right]$$

shows that while it is quadratic close to $x_i-x_j=0$ it penalises all large discrepancies to the same degree. Thus the prior avoids the problems of "over-penalising" very large discrepancies which might occur as a result of a natural boundary.

A simpler problem

We shall now simplify the problem a little and describe the reconstruction procedure. We assume that the underlying levels are known to be $m,2m,\ldots,km$ where m and k are known integers. This simplifies the problem in two ways; firstly the underlying levels are known and we can produce a reconstruction in which every pixel takes one of these colours. Also the underlying levels are equally spaced, we will see later that this assists us in the choice of the parameter α. The ICM method is used to find a local maximum of the posterior distribution, choosing at each stage the colouring of pixel i which minimises

$$\frac{1}{2\sigma^2}(y_i-x_i)^2 + \beta\sum_{j\in\partial_i}\left[1-\frac{1}{1+\alpha(x_j-x_i)^2}\right]$$

Once a reconstruction has been obtained with values $x_i\in\{m,m+1,\ldots,km\}$, each pixel is assigned the underlying grey level from the set $\{m,2m,\ldots,km\}$ which its value is closest to, i.e. if pixel i has colouring x_i in the reconstruction then it is assigned level pm such that $|pm-x_i|$ is at its minimum over $p=1,\ldots,k$. This is called the "binning" stage. As is common in most applications of ICM, we use the closest mean classifier as the initial estimate of the scene.

Choosing the parameters

The parameter β is used in the same way as for binary reconstruction and is not discussed here. In the examples that follow the second order neighbours have been down weighted by a factor of $1/\sqrt{2}$ as in previous experiments. The choice of α is important. It determines the amount of variation that is permitted within regions. Clearly we desire a value which allows maximum variation within regions but is not detrimental to the reconstruction process at the region edges. Consider the effect of the prior on a pixel which has half of its neighbours coloured pm $(0\le p<k)$ and the remainder coloured $(p+1)m$. The prior should favour either pm or $(p+1)m$ against $pm+m/2$. This characteristic is achieved if

$$2\left[1-\frac{1}{1+\alpha(m/2)^2}\right] > \left[1-\frac{1}{1+\alpha m^2}\right]$$

which occurs if and only if

$$\alpha > \frac{2}{m^2}.$$

There are arguments which suggest that the prior should be much more likely to assign one of the underlying levels pm or $(p+1)m$ in the case of the pixel that we have described above. This corresponds to choosing a value of α much larger than $2/m^2$ and restricts the variability within regions. The danger of this is that small clusters of pixels appear in the reconstruction inside regions which will be binned to a different grey level. Because ICM chooses the modal value at each stage of the process, the choice of α is not as crucial as it would be if simulated annealing was used. We have found that using $\alpha \approx (2/m^2) \times 1.1$ gives good results.

An example

Figure 4.24 shows a 128 by 128 4 colour scene in which the pixels take values 0, 11, 22 and 33. The true value for each pixel is perturbed by adding independent Gaussian noise with mean 0 and variance 20. In order that the variance of the record may be compared with binary and other examples we introduce the *signal to noise ratio* which is the difference between the mean levels divided by the standard deviation of the noise. For this example the signal to noise ratio is $11/\sqrt{20}=2.46$. Figure 4.25 shows the closest mean classifier for the record which is used as an initial estimate of the true scene. The unbinned ICM reconstruction obtained using $\alpha=0.02$ and $\beta=1.0$ is shown in Figure 4.26. Figure 4.27 shows the corresponding binned reconstruction. These results are far superior to those obtained using other priors. The priors which we experimented with were

$$p(x) \propto \exp\{-\frac{\beta}{2} \sum_{i=1}^{n} \sum_{j \in \partial_i} I(x_i \neq x_j)\}$$

which penalises all discrepancies equally and

$$p(x) \propto \exp\{-\frac{\beta}{2} \sum_{i=1}^{n} \sum_{j \in \partial_i} (x_i - x_j)^2\}.$$

which penalises the difference between pixel colourings and the mean colouring of their neighbours.

The original problem

We now return to the original problem, where the underlying levels are unknown. The method that we have described may still be applied but there are

Figure 4.24

The true scene

Size 128×128

Levels: 0, 11, 22, 33. R=33

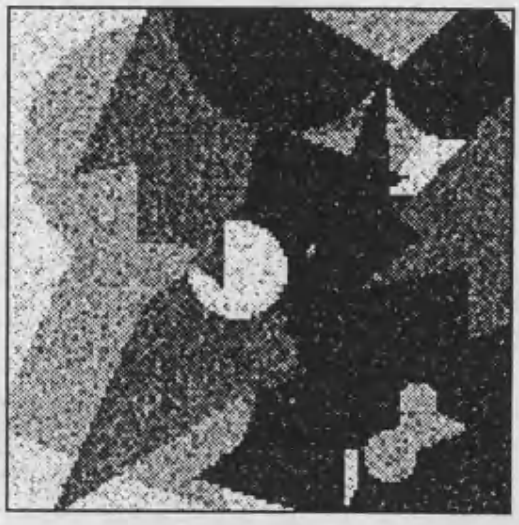


Figure 4.25

The closest mean classifier

Number of pixels misclassified: 3159

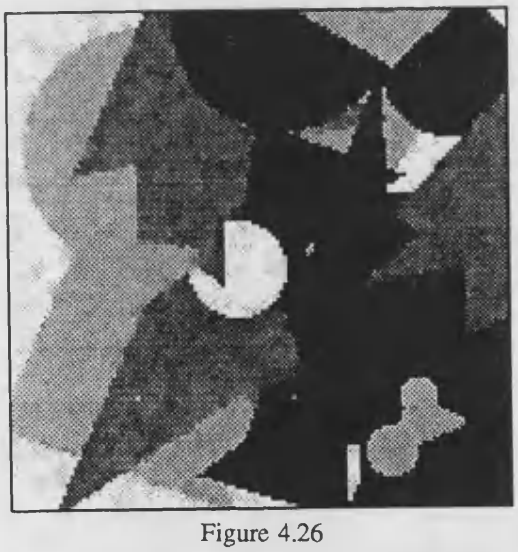


Figure 4.26

Reconstruction technique: ICM, known levels (unbinned)

Parameters: $\beta_1 = 1.0$

$\beta_2 = \beta_1/\sqrt{2}$

α=0.02

Initial estimate: Closest mean classifier

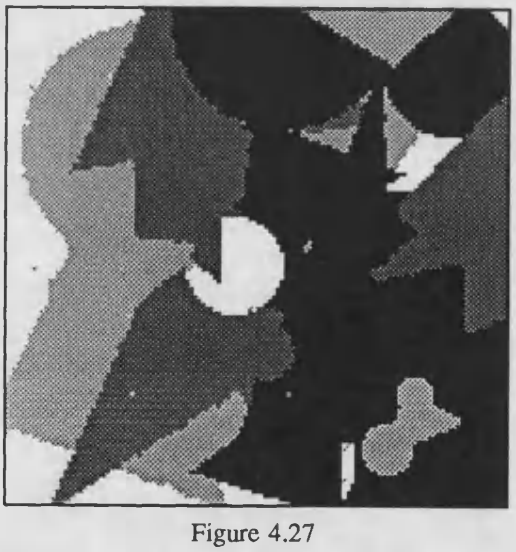


Figure 4.27

Reconstruction technique: ICM, known levels (binned)

Parameters: $\beta_1 = 1.0$

$\beta_2 = \beta_1/\sqrt{2}$

α=0.02

Initial estimate: Closest mean classifier

Number of pixels misclassified : 134

two additional considerations. First, α must be selected in a different way, since the choice depends on the unknown separation of the underlying levels. Also, the values which are used as bins in the final stage of the process are unknown and must be estimated. Reliable estimates of these values may be obtained if an adaptive approach is adopted and we shall describe the procedure by means of an example. Figure 4.28 shows a 128 by 128 scene in which each pixel takes one of the values 8, 16 29 or 55. The range of possible pixel colourings is [0,63]. Gaussian noise with variance 50 is added to the true value for each pixel and the closest mean classifier is shown in Figure 4.29. Note that the underlying levels are unknown and so in the closest mean classifier each pixel is assigned the value in [0,63] which is closest to its record. We first obtained a reconstruction using α=0.01 in the prior (4.4.1), a value which corresponds to a separation of about 15 grey levels between the underlying levels. Although this initial estimate was not a good one, the unbinned reconstruction obtained is fairly good and is shown in Figure 4.30. The fitted values from this reconstruction are shown in the histogram in Figure 4.31 which indicates that the separation between the underlying levels is as small as 8. For a separation of 7 to 8 grey levels in the underlying scene we would choose to use α=0.038 and Figure 4.32 shows the unbinned reconstruction obtained using this value. The histogram of the fitted values is shown in Figure 4.33 which suggests that the bins are at the levels 9, 16, 29 and 55. Notice that the peaks are much better defined in the histogram in Figure 4.33 than in Figure 4.31. This is because fewer pixels are classified as being between levels. The binned reconstruction is shown in Figure 4.34.

4.4.2 Using the cascade algorithm

The ICM method that we have described works well in the examples that we have shown. Where the variance of the noise is larger, reconstructions obtained using the ICM method are unsatisfactory. Recall that this was also true for the binary examples that were shown earlier in this chapter. We shall again use the cascade algorithm to reduce the the variance of the data and to assist in the reconstruction process. Figure 4.35 shows a true scene where each pixel takes a value 10, 25, 40 or 55 and the range of possible values is [0,63,]. Gaussian noise with variance 900 is added to the true value at each pixel and the closest mean classifier is shown in Figure 4.36. This variance corresponds to a signal to

Figure 4.28

The true scene
Size 128 × 128
Levels: 8, 16, 29, 55. R=65

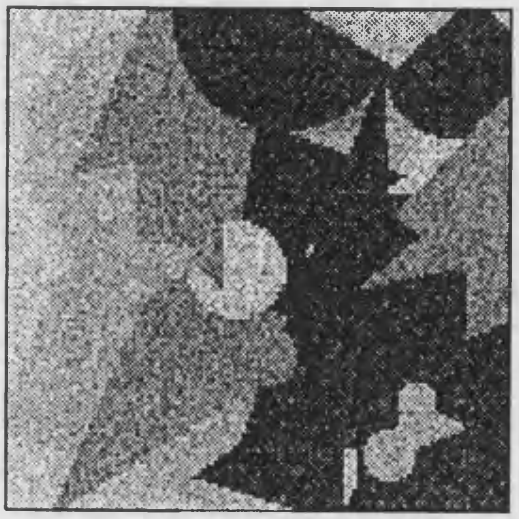


Figure 4.29

The closest mean classifier

Figure 4.30

Reconstruction technique: ICM, unknown levels
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
α=0.01

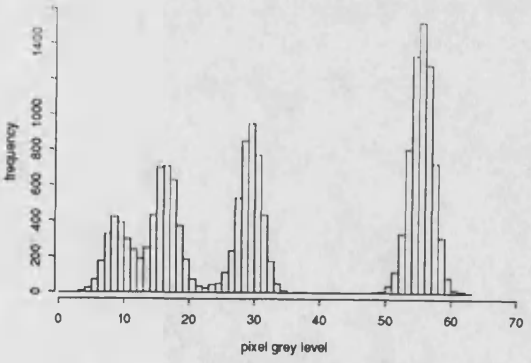


Figure 4.31

Histogram plot of fitted values for Figure 4.30

Figure 4.32
Reconstruction technique: ICM, unknown levels
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
α=0.038

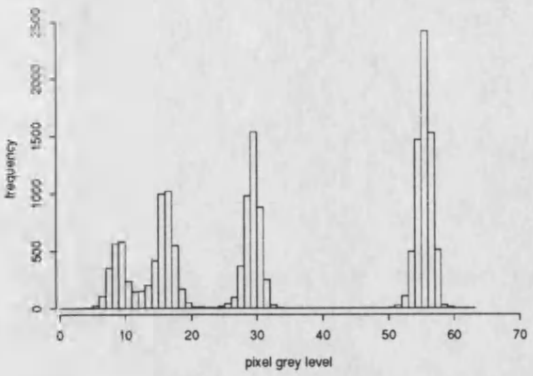


Figure 4.33
Histogram plot of fitted values for Figure 4.32

Figure 4.34
Reconstruction technique: ICM, estimated levels (binned)
Parameters: as for Figure 4.32
Estimated levels: 9, 16, 29, 55.

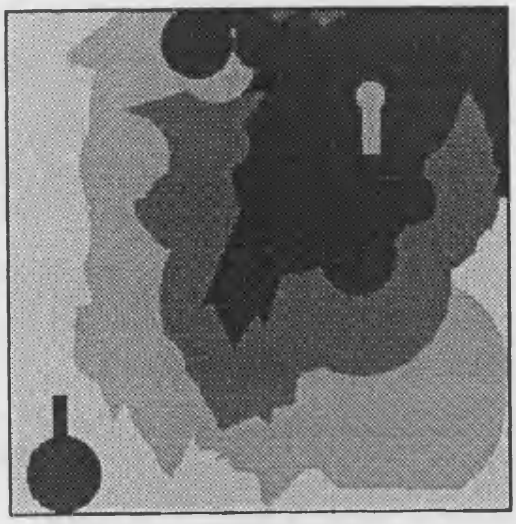


Figure 4.35

The true scene
Size 256 × 256
Levels: 10, 25, 40, 55. R=65

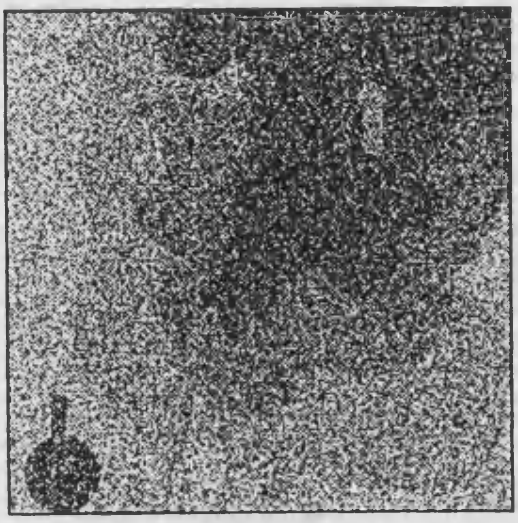


Figure 4.36

The closest mean classifier
Number of pixels misclassified : 39147

Figure 4.37

Reconstruction technique: ICM, known levels
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
α=0.01

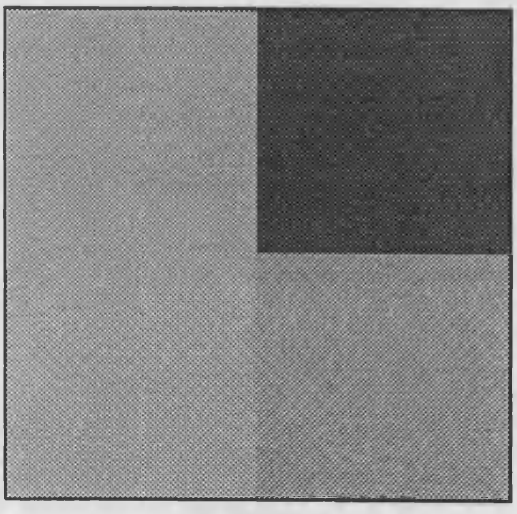


Figure 4.38

Reconstruction technique: ICM, unknown levels
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
α=0.01
Initial estimate: unbinned 1 × 1 cascade
Dimensions: 2 × 2
Record: 256 × 256 aggregated 7 times

noise ratio of 0.5. The underlying levels are unknown and so in the closest mean classifier each pixel is assigned the value in [0,63] which is closest to its record. The ICM reconstruction shown in Figure 4.37 was obtained with the underlying levels known but is still of fairly poor quality. However we can incorporate the ICM technique into the cascade algorithm in a similar way as for binary images to obtain excellent reconstructions for this record. Figures 4.38 to 4.45 show the cascade reconstructions at different stages of the process which were obtained using α=0.01. This estimate of α was obtained by first executing the cascade using a sensible guess for α; the histogram of fitted values was inspected at the final stage and interpreted in the same way as for the previous example. The value α=0.01 corresponds to a separation of 14 to 15 between the underlying levels. The bins which were eventually used were 11, 25, 40 and 55 and Figure 4.45 shows the binned reconstruction. During the cascade process the unbinned reconstructions are used as an initial estimate for the next level. Initially we experimented with using the binned reconstructions at each level as the initial estimates but the results were poor.

In the binary example we considered the possibility of an early stopping criterion for the algorithm. The criterion may also be applied to the grey level cascade and we observe that only 1.2% of the pixels shown in the 256 by 256 reconstruction have changed colour at that level of the cascade.

4.5 Summary

In this chapter we have introduced an algorithm which enables the large scale features of an image to be successfully reconstructed even when the noise level is high. The cascade algorithm successfully uses aggregation to exploit the dependence of the local updating procedure ICM on the initial estimate of the scene.

The technique that we have described for grey level reconstruction requires intervention at several stages. There may be difficulties with the interpretation of the histogram and the choice of α, but in the experiments that we have conducted these tasks have not proved arduous. The examples that we have shown illustrate the potential of the technique to deal with realistic problems.

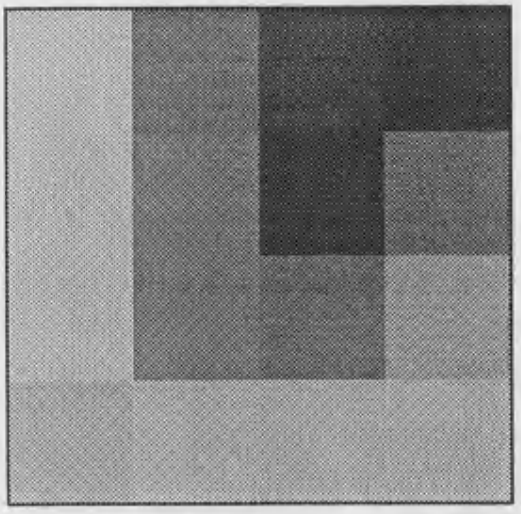


Figure 4.39

Reconstruction technique: ICM, unknown levels

Parameters: $\beta_1 = 1.0$

$\beta_2 = \beta_1/\sqrt{2}$

$\alpha = 0.01$

Initial estimate: unbinned 2 × 2 cascade (Fig 4.38)

Dimensions: 4 × 4

Record: 256 × 256 aggregated six times

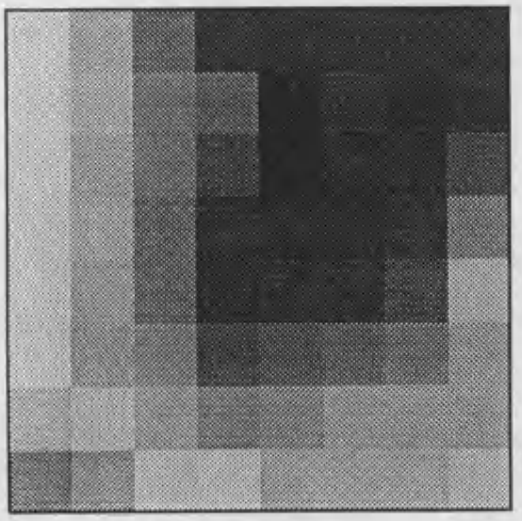


Figure 4.40

Reconstruction technique: ICM, unknown levels

Parameters: $\beta_1 = 1.0$

$\beta_2 = \beta_1/\sqrt{2}$

$\alpha = 0.01$

Initial estimate: unbinned 4 × 4 cascade (Fig 4.39)

Dimensions: 8 × 8

Record: 256 × 256 aggregated five times

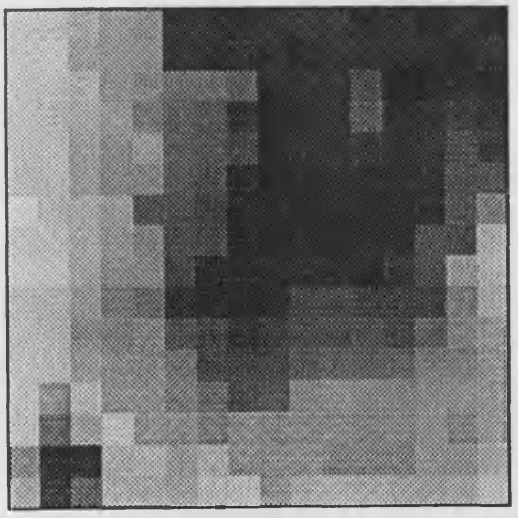


Figure 4.41

Reconstruction technique: ICM, unknown levels

Parameters: $\beta_1 = 1.0$

$\beta_2 = \beta_1/\sqrt{2}$

$\alpha = 0.01$

Initial estimate: unbinned 8 × 8 cascade (Fig 4.40)

Dimensions: 16 × 16

Record: 256 × 256 aggregated four times

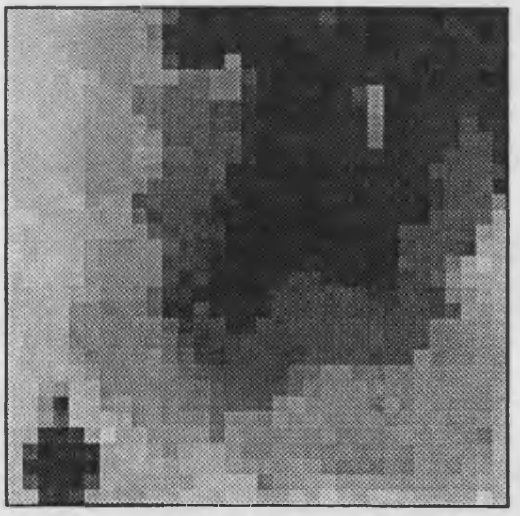


Figure 4.42

Reconstruction technique: ICM, unknown levels

Parameters: $\beta_1 = 1.0$

$\beta_2 = \beta_1/\sqrt{2}$

$\alpha = 0.01$

Initial estimate: unbinned 16 × 16 cascade (Fig 4.41)

Dimensions: 32 × 32

Record: 256 × 256 aggregated three times

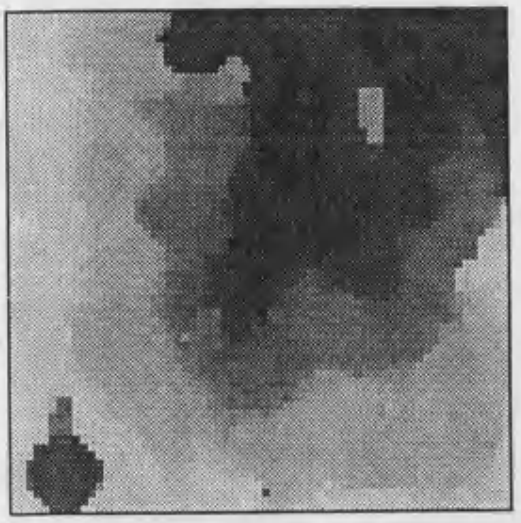


Figure 4.43

Reconstruction technique: ICM, unknown levels

Parameters: $\beta_1 = 1.0$

$\beta_2 = \beta_1/\sqrt{2}$

$\alpha = 0.01$

Initial estimate: unbinned 32 × 32 cascade (Fig 4.42)

Dimensions: 64 × 64

Record: 256 × 256 aggregated twice

Figure 4.44

Reconstruction technique: ICM, unknown levels

Parameters: $\beta_1 = 1.0$

$\beta_2 = \beta_1/\sqrt{2}$

$\alpha = 0.01$

Initial estimate: unbinned 64 × 64 cascade (Fig 4.43)

Dimensions: 128 × 128

Record: 256 × 256 aggregated once

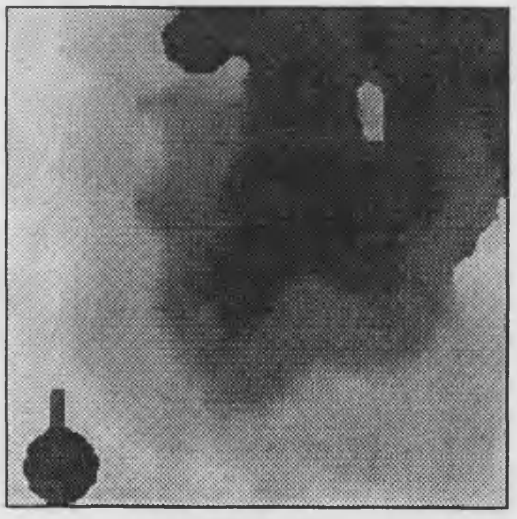


Figure 4.45

Reconstruction technique: ICM, unknown levels

Parameters: $\beta_1 = 1.0$

$\beta_2 = \beta_1/\sqrt{2}$

$\alpha = 0.01$

Initial estimate: unbinned 128 × 128 cascade (Fig 4.44)

Dimensions: 256 × 256

Record: original 256 × 256 record

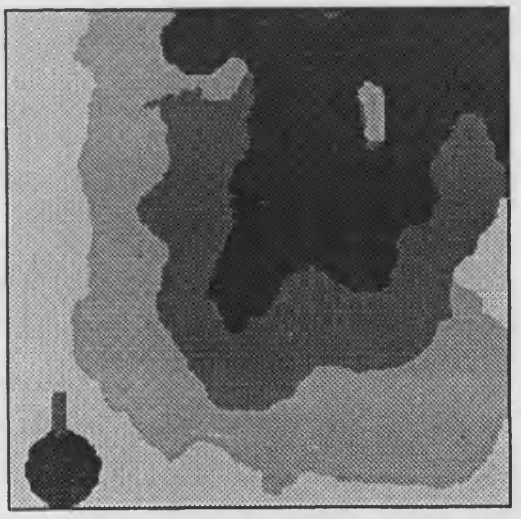


Figure 4.46

Reconstruction technique: ICM, unknown levels binned to estimated bins

Parameters: as for Figure 4.45

Estimated levels: 11, 25, 40, 55.

Chapter 5: Subpixel refinement

5.1 Introduction

So far the images that we have considered have contained only pixels which are wholly one colour. In practical applications this will rarely be the case, for example, LANDSAT (land satellite) data, used to classify rock types or land use, may be collected over very large areas of land. Switzer (1983) deals specifically with LANDSAT data and uses an example where each pixel measures about an acre in size; the problems of classifying pixels which are on the boundaries of class types are recognised. In the context of edge detection Huertas and Medioni (1986) extend masking methods to attain subpixel precision. Single photon emission computer tomography (SPECT) and positron emission tomography (PET) are applications in which the data are indirectly observed and the orientation and size of the pixel grid may be chosen freely. This will often lead to many split pixels, i.e. pixels which contain more than one region type. Jennison (1986) extends the method of Iterated Conditional Modes to a subpixel level: we shall describe this technique in detail and develop it to greater generality.

5.2 Quarter-pixel Iterated Conditional Modes

This technique was developed for binary imaging although there are possibilities for extending the method to handle more classes. At this point we must introduce more notation since the true value x_i^* at pixel i no longer takes values 0 or 1. We use z_i to denote the colouring which uniquely defines the configuration of the black and white areas within pixel i, $z=\{z_i\}$ defines a colouring of the whole region and we use $pn(z_i)$ to denote the proportion of pixel i which is coloured black when the colouring is z_i. The true colouring of pixel i is denoted by z_i^*. We assume that the record y_i at pixel i is Gaussian with mean $pn(z_i^*)$ and variance σ^2. The problem is to estimate $z^*=\{z_i^*\}$ from the set of records, y_i.

Jennison (1986) considers reconstructions in which quarter pixels are coloured individually. Each pixel is divided into 2×2 square quarter pixels and a MRF model is constructed which penalises discrepancies between quarter pixel neighbours, i.e. quarter pixels are treated in the same way as full size pixels have previously been treated. Notice that each quarter pixel has neighbours which are

both inside and outside of its larger parent pixel. The prior model which is used is

$$p(x) \propto \exp -\frac{\beta}{2} \sum_{i=1}^{n} \sum_{j=1}^{2} \sum_{k=1}^{2} Z(x_{ijk}) \tag{5.1}$$

where x_{ijk} corresponds to quarter pixel jk of pixel i and takes the value 0 or 1; $Z(x_{ijk})$ is the number of discrepant quarter pixel neighbours of quarter pixel jk of pixel i. The factor 1/2 is included as each neighbour pairing will be counted twice. The pixels are updated in a similar way to ICM, at each pixel the colouring of its 4 quarter pixels is chosen to minimise the contribution to

$$\sum_{i=1}^{n} \frac{1}{2\sigma^2} \left[y_i - \frac{1}{4} \sum_{j=1}^{2} \sum_{k=1}^{2} x_{ijk} \right]^2 + \frac{\beta}{2} \sum_{i=1}^{n} \sum_{j=1}^{2} \sum_{k=1}^{2} Z(x_{ijk}). \tag{5.2}$$

The initial colouring at each subpixel is determined from an ICM reconstruction which is first obtained in the full pixel grid. As with the cascade algorithm no attempt is made to reconcile the models at the two levels.

An example

Figure 5.1 shows the true scene. A 32 by 32 grid is superimposed on the image and the proportion $pn(z_i)$ of black in each pixel is recorded. The record y_i at pixel i is distributed as Gaussian with mean $pn(z_i)$ and variance 0.04. The ICM reconstruction is shown in Figure 5.2 and this is used as a starting point for the quarter pixel technique which gives the reconstruction shown in Figure 5.3. The circle in the top left of the image has been well reconstructed. The star shape contains points which are not even a pixel in width and it is not surprising that this shape has not been well reconstructed.

In general we have obtained good results using the same value of β at both grid levels, i.e. for both the initial full pixel ICM and the subsequent quarter pixel technique. When the signal to noise ratio is high both methods are fairly robust and the choice of the parameter will not be crucial.

5.3 Further refinement

The success of Jennison's method suggests that further subpixel refinement may be beneficial. The obvious progression would be to consider the $m \times m$ breakdown of pixels in which each pixel is divided into m^2 square pixels and

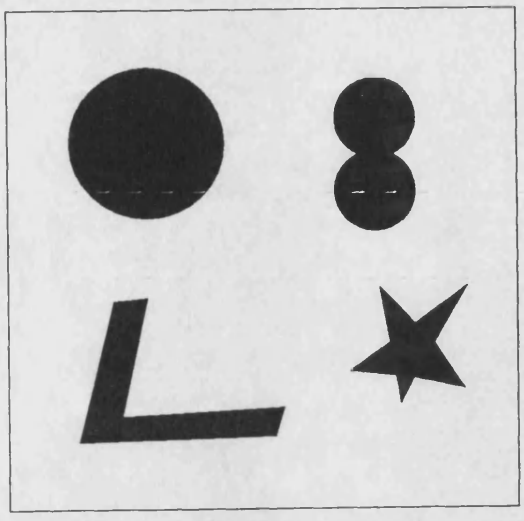


Figure 5.1
The true scene

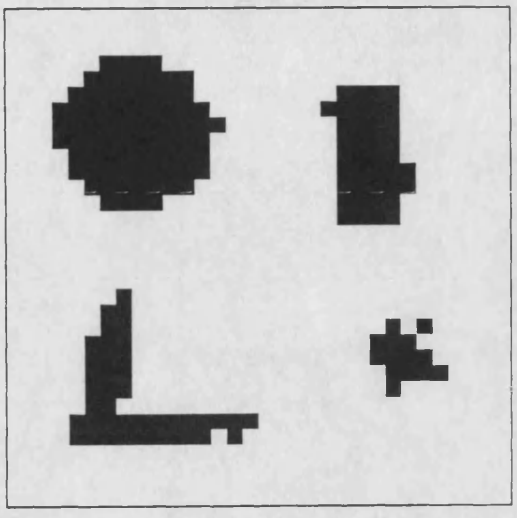


Figure 5.2
Reconstruction technique: ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: Closest mean classifier

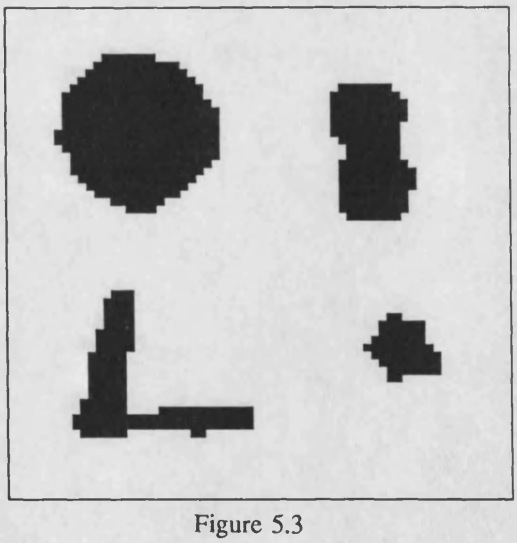


Figure 5.3
Reconstruction technique: Jennison's quarter pixel technique
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: ICM reconstruction (Figure 5.2)

each of these subpixels is coloured individually. Denoting by x_{ijk} the colouring of subpixel jk of pixel i, $\{x_{ijk}\}$ is chosen to minimise

$$\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \frac{1}{m^2}\sum_{j=1}^{m}\sum_{k=1}^{m}x_{ijk})^2 + \frac{\beta}{2}\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{m}Z(x_{ijk}). \qquad (5.3)$$

If all m^2 subpixels of one pixel are to be updated in one step of the ICM method, the number of different colourings of a pixel grows exponentially with the number of subpixels and the minimisation will become computationally prohibitive. We shall not consider this approach further. Instead we consider the limit of the process, in which an arbitrary colouring of each pixel is allowed. Rather than specify a MRF model for the true scene in the continuum, we interpret the minimisation of the objective function as a form of penalised maximum likelihood. In the pixel model, the second term of the objective function is, approximately, a multiple of the total boundary length in the image. Thus, an analogous objective function for a general reconstruction, z, is

$$\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - pn(z_i))^2 + \gamma L(z) \qquad (5.4)$$

where $L(z)$ is the total edge length in scene z and γ is a fixed constant. All pixels have dimensions 1×1. An advantage of using edge length as a measure is that it is rotationally invariant, a property that is impossible to obtain using pixels grids, however fine.

5.4 Finding subpixel colourings

In this section we investigate the form of the scene $\{z_i\}$ which minimises the objective function given by (5.4) and we propose an approach which may be used to find a local minimum.

5.4.1 Theorems relating to the form of the global solution

By considering the form of the solution locally, properties of the global solution may be discovered. In any solution the areas of black and white are separated by edges. We refer to sections of edge as paths and present two important theorems concerning the properties of paths in the global solution. We shall show:

1 Any path inside a pixel separating regions of colour is an arc of a circle or a straight line.

2 There are no discontinuities in the direction of the tangent to the path.

Thus, the solution might be of the form shown in Figure 5.4

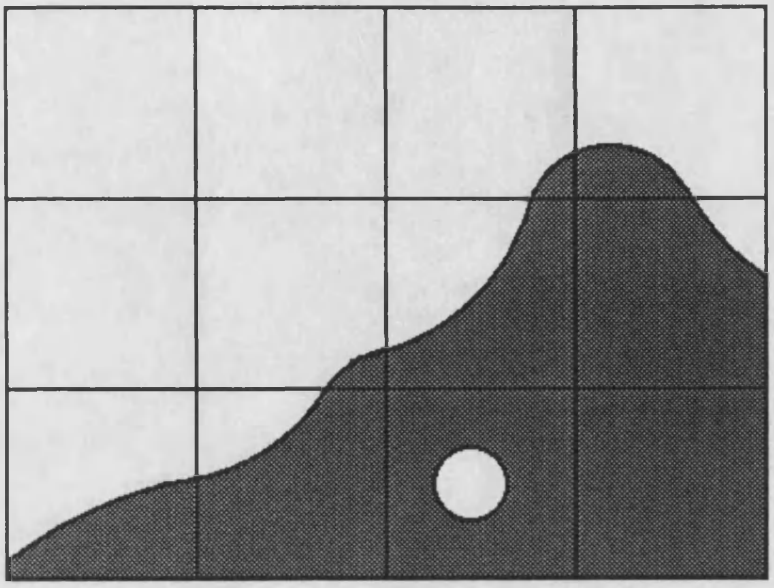


Figure 5.4

Theorem 1

Suppose that the solution contains a path, between two points a and b, which lies wholly inside a pixel, except possibly at the points a and b which may lie on the pixel edge. This path must be either a straight line or the arc of a circle.

Proof :

Suppose a and b are interior points of the pixel. Let c be any point on the path ab. We shall show how to construct a small interval on the path around c in which the path is a straight line or the arc of a circle. We can consider the set of such intervals which may be constructed on the path ab which provides an open cover for the path ab and, since the path ab is closed and bounded, by the Heine-Borel Theorem there is a finite subcover of the path for which each overlapping interval is a straight line or the arc of a circle. It follows that the whole of the path ab must be either a straight line or the arc of a circle. If a and b are not interior points of the pixel then, by the above argument, for any points a' and b'

on the path ab, the path $a'b'$ is a straight line or the arc of a circle; taking the limit as $a' \rightarrow a$ and $b' \rightarrow b$ we see that the path ab must also be a straight line or the arc of a circle.

It remains to show that for every point c on the path ab we may construct a small interval on the path ab which is a straight line or the arc of a circle.

If there are points d and e, one either side of c, such that the path between d and e is a straight line then we take the interval (d,e).

If the path at c is not locally a straight line, it is easily seen that we can find points d and e, one on either side of c, such the path P between d and e is all on one side of the straight line de. For example, if the path lies to one side of the tangent at c in a neighbourhood of c, as in Figure 5.5, then we may translate the tangent at c by some distance ε, giving the points d and e as shown in Figure 5.5.

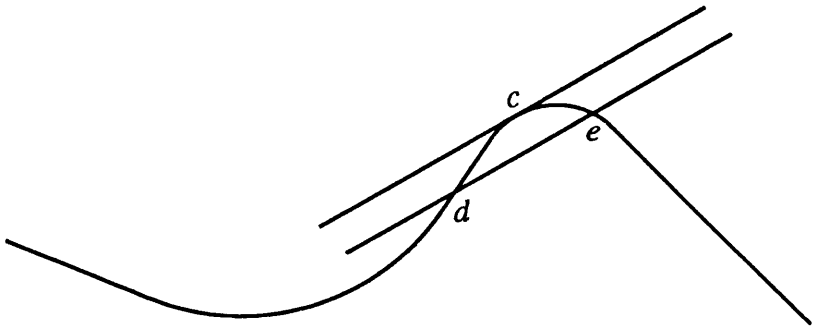


Figure 5.5

If the path at c lies above the tangent at c to one side of c and below this tangent to the other side, as shown in Figure 5.6, then the points d and e may be found by rotating the tangent by ε radians, giving the line L_1, and then translating L_1 by distance ε to give the line L_2 and the points d and e. For both of these cases the distance ε must be chosen to be sufficiently small so that the points d and e occur between the points a and b. There is an additional constraint which must also be satisfied and this is explained later in the proof.

Having found points d and e we now show that the path between d and e must form the arc of a circle. Suppose that the path P between d and e is not the arc of a circle (see Figure 5.7). Let Z be the area enclosed by the path P and the straight line de. Because P is part of a solution it has minimum length of all paths between d and e enclosing area Z with the line de. We construct the unique circle

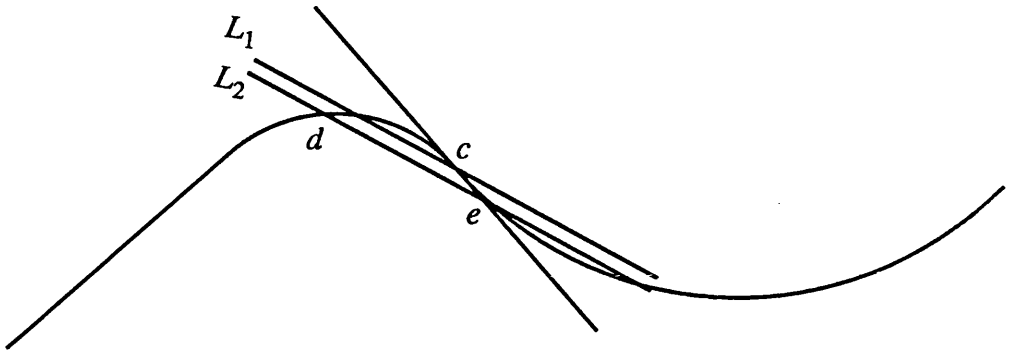


Figure 5.6

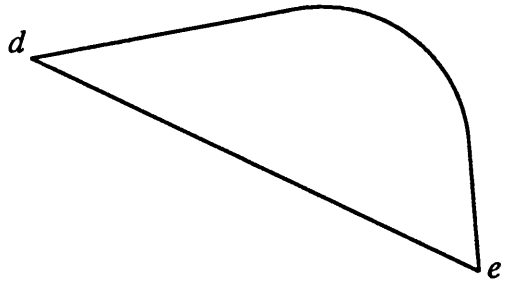


Figure 5.7

and let the total area of the circle be *Y*. Consider the path formed when the path *P* is combined with part of the circle *C* as shown in Figure 5.8b.

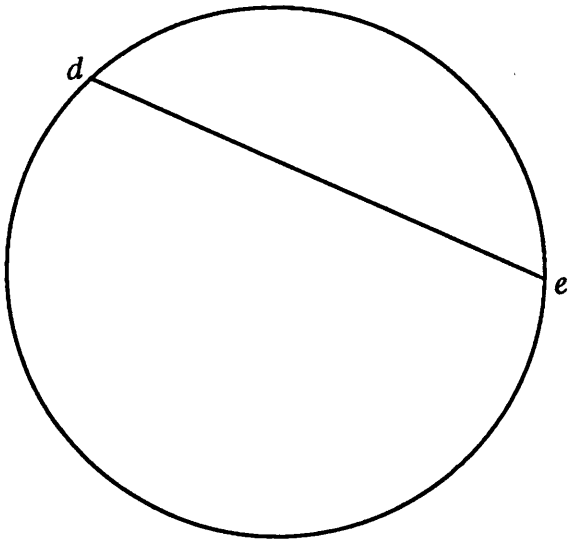


Figure 5.8a

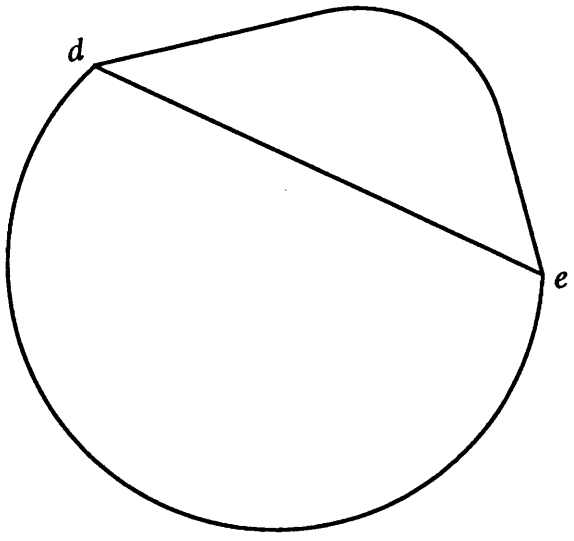


Figure 5.8b

This path also encloses total area *Y*. Pars (1962) proves that the object which has

minimum boundary for coverage of a given area is a circle. It follows that the circle C has a smaller perimeter than the path in Figure 5.8b. Since the only difference between these paths occurs between d and e it follows that the path P is inferior to the path given by the arc of the circle C.

Although we have constructed a path which is superior to the original the new path must also satisfy an additional constraint. It must remain inside the pixel and may not interfere with other colourings inside the pixel. This constraint is satisfied if ε is chosen sufficiently small so that the path between d and e, which encloses the same area with the straight line de as the path P, remains inside the pixel and may not interfere with other colourings inside the pixel. That the constructed path is superior to the original is a contradiction and so the original path P cannot form part of a solution. Thus, our supposition that the path between d and e was not the arc of a circle is shown to be false and this completes the proof.

Theorem 2

There are no discontinuities in the direction of the tangent to the path.

Proof :

Suppose a solution contains a discontinuity in the direction of the tangent to the path, as shown in Figure 5.9.

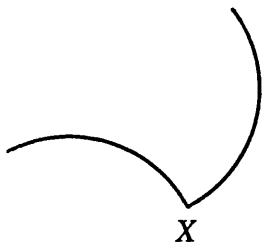


Figure 5.9

We consider the effect on the objective function of a small change in the path close to the discontinuity at X. Construct the points a and b on the two lines such that both are distance ε from the point X (see Figure 5.10a). Let the path between a and b change to a straight line as shown in Figure 5.10b. The reduction in the length of the path is of order ε. The first term of the objective function may involve as many as 4 pixels (since the point X may lie on the vertex of 4 pixels),

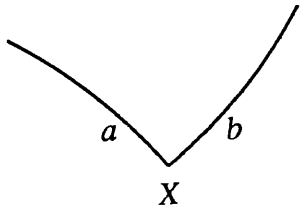


Figure 5.10a

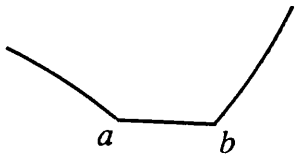


Figure 5.10b

but for each pixel there will be a change in the objective function of order ε^2. Thus, for sufficiently small ε there will be a decrease in the overall value of the objective function as the result of this change. We have constructed a path which is superior to the original and thus the original path may not form part of the solution. So the solution may not contain discontinuities in the direction of the tangent. □

5.4.2 Other results

Theorems 1 and 2 are crucial to the understanding of the nature of the global solution. There are some more results which also increase this understanding although they refer to particular cases. Rather than present formal proofs for each of these, an informal justification is given.

(a) If a pixel in the solution has a boundary which is all one colour then the pixel will either:

- be all one colour,
- contain a circle of opposite colour (this follows from Theorem 1),

or

- contain a shape made up of straight lines along pixel edges joined by quarter arcs of circles, as shown in Figure 5.11; the quarter arcs having a common radius. (This follows from Theorems 1 and 2 combined with some more detailed analysis involving symmetry.)

(b) Where a pixel in the solution has a neighbouring boundary as shown in Figure 5.12a the path between a and b will be the arc of a circle (Figure 5.12b) or will be made up of straight lines on the pixel boundary joined by arcs of circles which have common radius, (two possible solutions are shown in Figure 13).

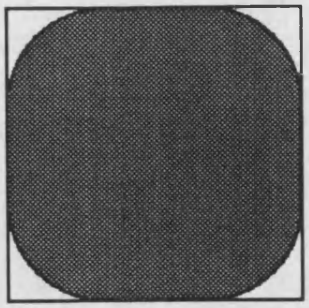


Figure 5.11

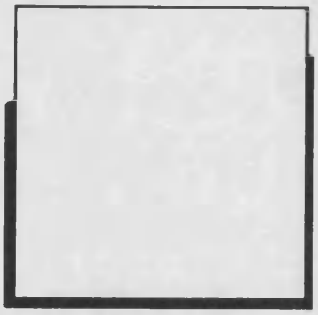


Figure 5.12a

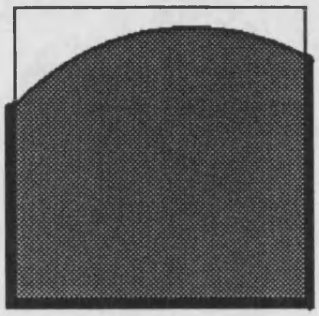
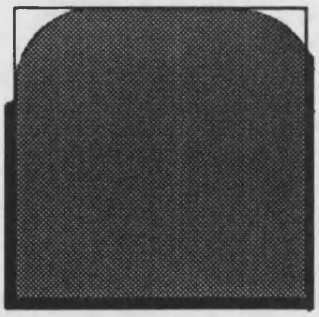


Figure 5.12b



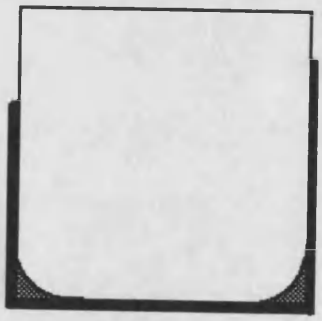


Figure 5.13

(c) A solution may contain several separate areas of the colour. However, if any of these areas may be moved, within the region, so that they touch another area of the same colour without increasing the value of the objective function then neither colouring may be a solution. This is illustrated in Figures 5.14a and 5.14b. If the colouring in Figure 5.14a is a solution, then so is the colouring in Figure 5.14b, but Theorems 1 and 2 demonstrate that Figure 5.14b cannot be a solution. This is a contradiction and therefore 5.14a cannot be a solution.

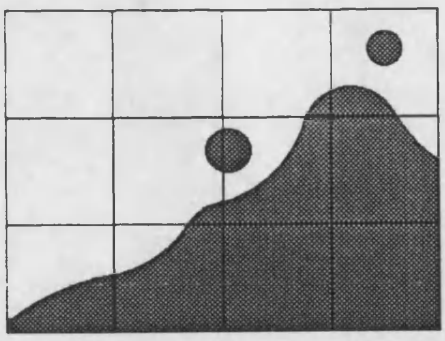


Figure 5.14a

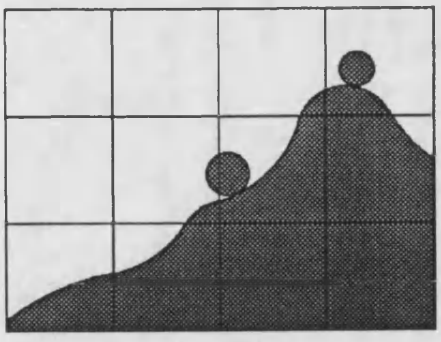


Figure 5.14b

(d) Pixels in the solution may have neighbours which have colourings which split the boundary into more than 2 sections of colour (see Figure 5.15a). The colourings of such pixels must still adhere to the conditions found in Theorems 1 and 2 and an example colouring is shown in Figure 5.15b.

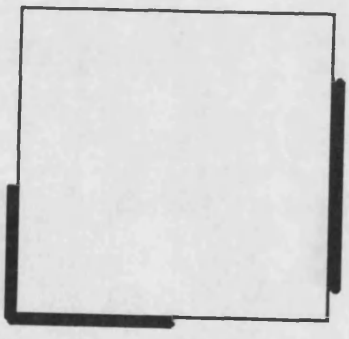


Figure 5.15a

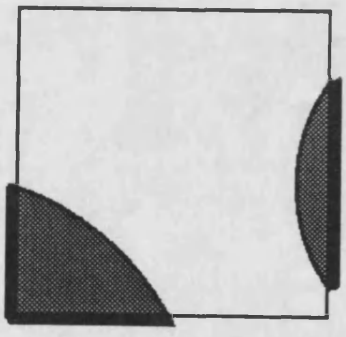


Figure 5.15b

5.4.3 Finding the solution

We have seen some of the properties of the global minimum of (5.4), and it is not clear that such a solution is directly attainable. Instead we may search for a local minimum using an ICM type approach. As part of an iterative scheme we may consider pixels individually, finding the best colouring for each given the colourings of all other pixels. The neighbour information which is used involves only the immediate neighbourhood, the colouring of the boundary of the pixel. As an example, we consider the configuration at pixel i as shown in Figure 5.16. The record for pixel i is y_i which is known; the lengths a and b to the points l and m are also known. The area D is a simple function of a and b and the area E

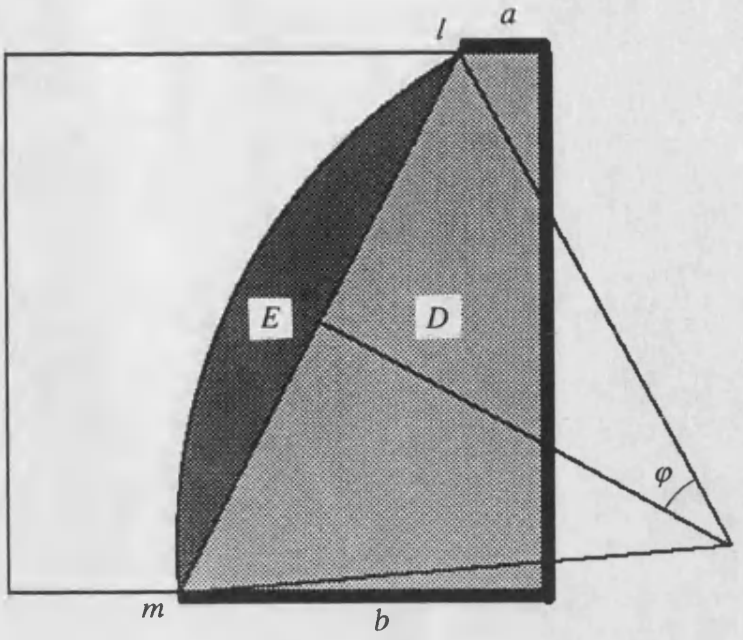


Figure 5.16

is a function of φ and the distance c between the points l and m. In fact

$$E = \frac{c^2}{4\sin^2\varphi}(\varphi - \sin\varphi\cos\varphi)$$

and the arc length is $c\varphi/\sin\varphi$.

So the contribution to the objective function from pixel i is

$$\frac{1}{2\sigma^2}(D+E-y_i)^2 + \frac{\gamma c\varphi}{\sin\varphi}$$

$$= \frac{1}{2\sigma^2}\left[D + \frac{c^2}{4\sin^2\varphi}(\varphi - \sin\varphi\cos\varphi) - y_i\right]^2 + \frac{\gamma c\varphi}{\sin\varphi} \qquad (5.5)$$

For the configuration shown in Figure 5.16, $D=a+(b-a)/2$ and $c=\sqrt{1+(b-a)^2}$. For this and other cases the value of φ which minimises (5.5) may not be found directly as the expression may have two local minima. However, this value of φ may be found quite easily using numerical techniques.

This provides the optimal arc for this single pixel given the colouring of its neighbours. However, iteratively updating all of the pixels in this way will not necessarily lead to to a local minimum of (5.4) satisfying the conditions of Theorem 2. Figure 5.17 shows the arcs for a pair of neighbouring pixels each of which is optimal given the pixel's neighbours. The tangent of the path at the join P is discontinuous and so this configuration may not form part of a solution. If the point P is allowed to move on the common pixel edge then the arcs which

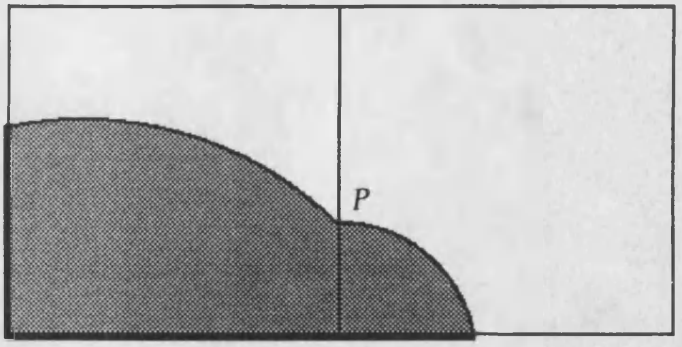


Figure 5.17

together minimise the contribution to the objective function, given the colourings at all other pixels, will join with a continuous tangent. Repeatedly updating the pixels in pairs around an area of colour will give a local minimum of the objective function which does satisfy Theorems 1 and 2. Because the objective function is reduced or remains constant at every updating, convergence, at least to a local minimum, is guaranteed. We shall not pursue this form of minimisation further, rather we shall consider an approximate method which avoids many of the complexities of this type of solution.

5.5 Implementation of a method for subpixel refinement

5.5.1 Introduction

In Section 5.4 we investigated the properties of the global minimum of (5.4). In this section we describe a method for obtaining an approximate solution which avoids many of the complications of an "arcs and lines" type solution, providing a practical technique for subpixel refinement. The technique is implemented and favourable results are obtained.

There are many difficulties encountered in the consideration of an "arcs and lines" minimisation. Firstly, for any region which is more than a few pixels in size the global minimum will, in general, be very difficult to find. If a local minimum is required then problems are caused by pixels whose boundaries are split into more than two sections by the colouring of their neighbours. Also, when a pixel needs a colouring which involves more than a single arc then the calculation of the correct colouring is complex. We impose constraints on the form that the solution may take to obtain a workable and practical method.

The constraints

- Each pixel may be either a single colour or separated into areas of different colour by a single straight line.
- Each pixel may contain at most one straight line.
- The image is regarded as a collection of line segments forming closed areas of colour, i.e. every line segment joins with another or meets with the boundary of the region.

The minimisation is local with pixel colourings updated in pairs by changing the point on the common edge at which a pair of line segments meet. Each area of black (an arbitrary choice) is considered in turn and the pixels which contain the edge of the area are registered, giving a list of pixels which form a route around the area. We define the *route* to be a collection of pixels which are ordered in a particular way, corresponding to a path (of pixels) which separates an area of black from an area of white. All routes must form closed circuits around areas of black and in some cases this may require that the route travels through imaginary pixels outside the region. Figure 5.18 shows an example of a solution together with the closed route showing which pixels were used in the procedure. The route outside the region is shown but updating only occurs inside the region. Once established, each route is considered in turn and the line segments within each pixel on that route are updated so that a local minimum is attained.

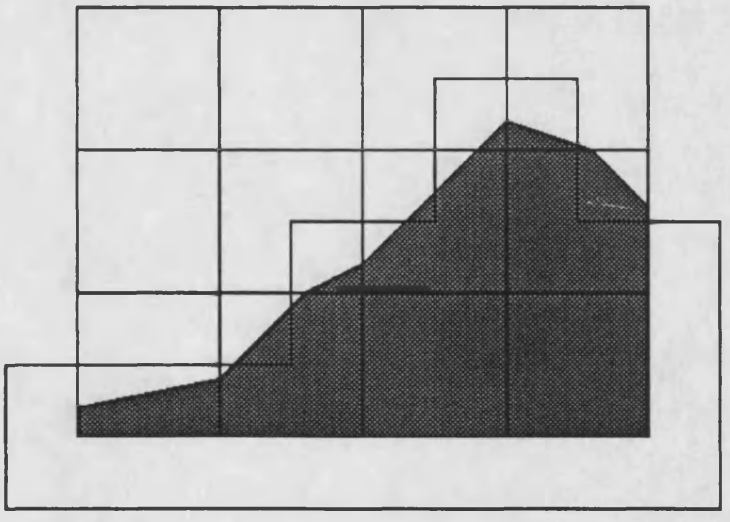


Figure 5.18

5.5.2 Finding the routes

An initial reconstruction is obtained using Jennison's quarter pixel method and this reconstruction is used to find the routes around areas of black in the region. Starting at a corner pixel the region is scanned in a raster fashion until a pixel is found which has quarter pixel neighbours which are not all the same colour. This pixel is labelled as the starting point for that route. Working around the area of black, each pixel is labelled as being on that route and its details are stored. At some stage the route will return to the starting pixel and it is then complete. The raster scan then continues but only unlabelled pixels are considered and other routes are found in a similar way. As we have already seen, the route will in some cases travel outside the region boundary. This is allowed because it ensures that all routes form closed circuits and simplifies the updating stage. We illustrate the route finding with an example. Figure 5.19a shows a quarter pixel reconstruction. From this the route is obtained and the pixels which lie on the route are shown in grey in Figure 5.19b. Initially the line segments join each other in the centre of the common pixel edge or an arbitrarily small distance (we used 0.05) from the ends of the common pixel edge, as shown in Figure 5.19c.

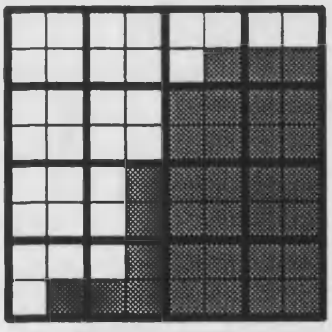


Figure 5.19a

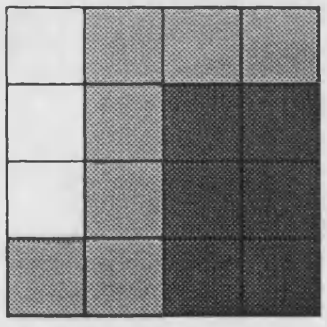


Figure 5.19b

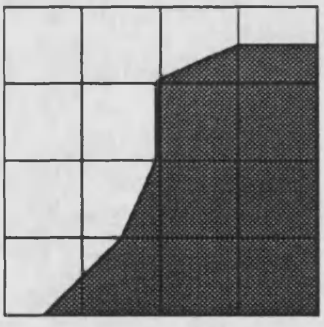


Figure 5.19c

Line segments are not permitted to join at a vertex of the pixel grid as this may lead to an undesirable local minimum in the early stages of updating.

For some quarter pixel configurations there are different routes that may be chosen. To avoid ambiguity and ensure that the route returns to its starting point, the pixels which are white are used whenever there is a choice. Each pixel may be part of only one route as only one line segment is permitted within each pixel in the reconstruction. Pixels in routes are labelled so that they are not used twice, either by the same route or other routes. Although this reduces the complexity of the procedure in the later stages, problems are encountered at the route finding

stage. Consider the quarter pixel reconstruction shown in Figure 5.20.

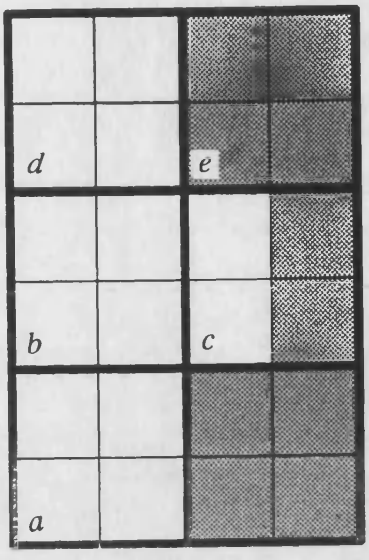


Figure 5.20

The algorithm that we have proposed would attempt to find the route *abcbd* and would fail since the pixel *b* cannot be used twice. Through the consideration of many special cases the program has been developed to handle some obstructive cases. For the configuration shown the route the modified algorithm would find the route *abced*.

5.5.3 Updating the line segments

Once the routes have been established, only pairs of pixels occurring on the routes need be considered for updating. Circuits of each route are repeated until convergence is achieved. At the start of the updating all the line segments join close to the ends or exactly in the centre of the common pixel edge, depending on the quarter pixel reconstruction from which the routes are found. Only rarely does the inaccuracy of the initial positionings of the line segments have a noticeable effect on the reconstruction obtained.

There are only 4 distinctly different ways in which the line segments may pass through a pair of pixels and these are shown in Figure 5.21 All other cases may be reduced to one of these by means of exchanging and/or inverting the pixels and their colours. We shall refer to the distance from the black end of the common pixel edge to the join of the line segments as the *cut value*, which takes a value in $[0,1]$. The contribution to the penalty for a particular cut value W is given by $g(W)$. The value of W which minimises $g(W)$ cannot be found directly

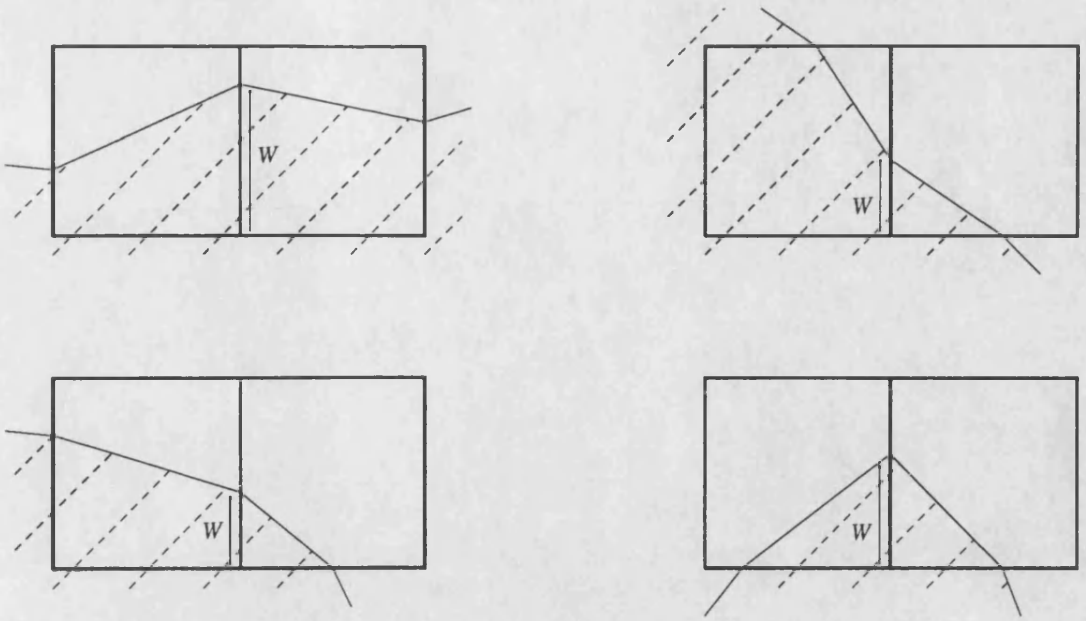


Figure 5.21

for any of the cases but an iterative numerical technique is used. An expression is found for W_s in terms of the parameters of the configuration and an estimate W_{s-1}. Starting from any sensible initial value W_0, accuracy to 3 decimal places was achieved after at most 4 iterations. In practice we take W_0 to be the value of W prior to this update. The derivation of the equations used for finding W are shown in Section 5.5.10.

Updating also takes place at the region edge, where the line segment which joins with the region edge may change its position. There are 2 distinctly different ways in which the line segment may pass through the pixel and these are shown in Figure 5.22.

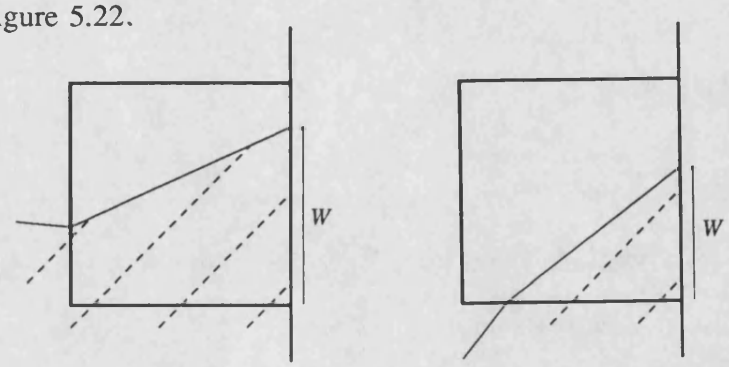


Figure 5.22

The equations for these cases are also shown in Section 5.5.10.

5.5.4 Choosing γ

The parameter γ is used in a similar way to the way that β is used in full pixel reconstruction. As is true with β there no single "correct" value of γ but we have found that using γ=4 gives good results in many different examples. Although this value of γ is recommended, its effectiveness and suitability will depend on the record and the nature of the true scene. Once the routes have been found from the quarter pixel reconstruction much of the processing has been completed and so, since it is only the updating stage of the process which relies on the value of γ, several different reconstructions may be quickly obtained for different values of this parameter. The effect of the choice of this parameter is illustrated using examples in Section 5.5.7.

5.5.5 Introducing an example

We shall use an example to demonstrate the effect of different levels of the parameter γ and to show the behaviour of the reconstruction at high and low noise levels. The example image shown in Figure 5.23 was first used by Jennison (1986) to demonstrate the quarter pixel technique. A 16 by 16 grid is superimposed on the image and the record for each pixel is obtained by adding Gaussian noise with mean 0 and variance σ^2 to the proportion of black in each pixel. We shall use β_1=1.0 and β_2=1/$\sqrt{2}$ for the full pixel and quarter pixel reconstructions in all of the examples.

Figure 5.24 shows the quarter pixel reconstruction obtained when σ^2=0.3. Given the level of the noise, this is a very good reconstruction. From this reconstruction the routes are obtained and Figure 5.25 shows the pixels (in grey) through which the routes travel. Figure 5.26 shows the line segment reconstruction for γ=4.

5.5.6 A development in the line fitting algorithm

The quality of the reconstruction that is obtained depends, to a certain extent, on the routes that the line segments take. Because these routes are fixed throughout the process other routes which might prove more successful are not considered. This is illustrated to a small degree by the example in Figure 5.26 where it can be seen that the pixels which have been used for the line segments are not the most suitable. We now propose a development in the line fitting

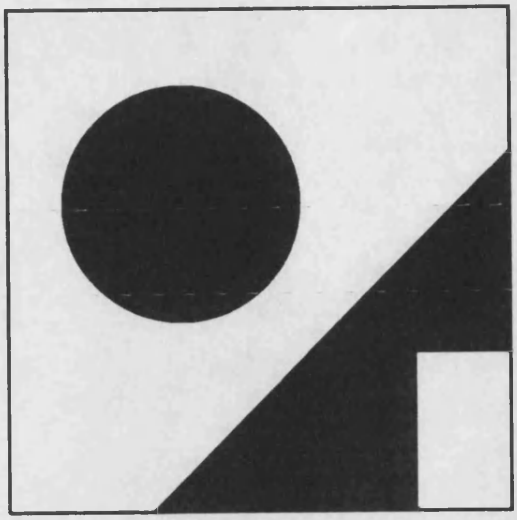


Figure 5.23
The true scene

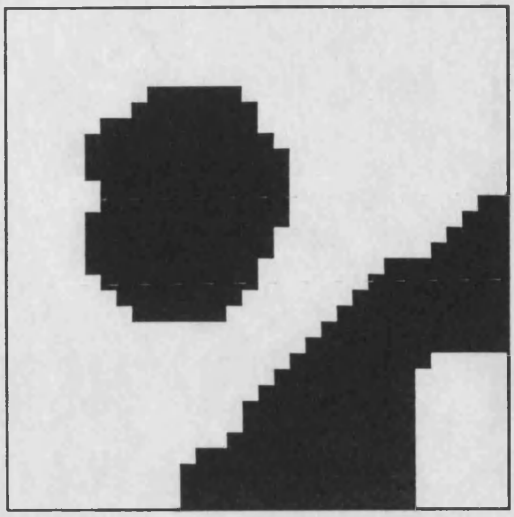


Figure 5.24
Reconstruction technique: Jennison's quarter pixel technique
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: ICM reconstruction

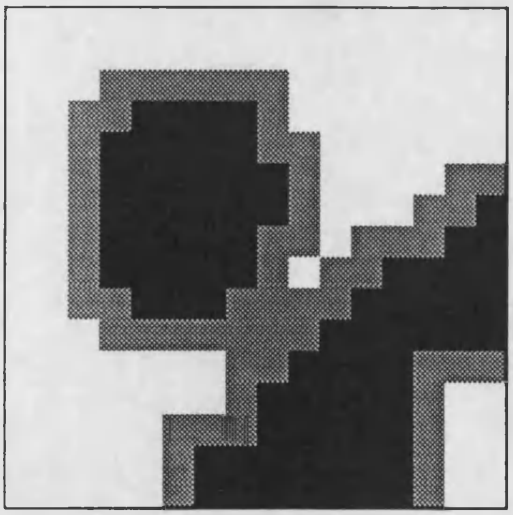


Figure 5.25
Pixels on the routes shown in grey

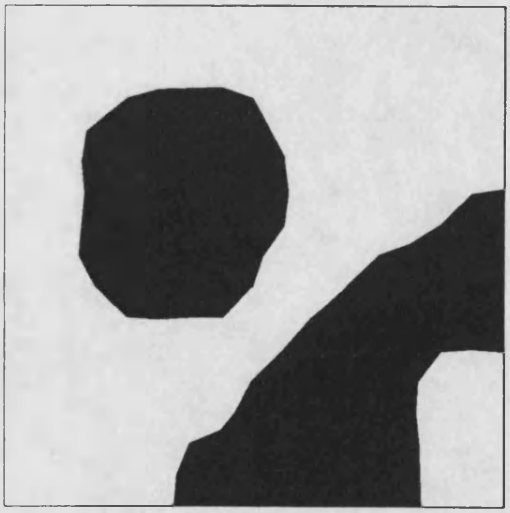


Figure 5.26
Reconstruction technique: Line fitting
Parameter: $\gamma = 4.0$
Initial routes obtained from Figure 5.24

algorithm which allows the routes to vary during the process.

Each time a line segment join is updated, alternative routes are considered. There are several distinctly different cases which must be treated separately and these are shown in Figure 5.27. Two different routes are shown in each case, and changes from one to the other are considered when appropriate.

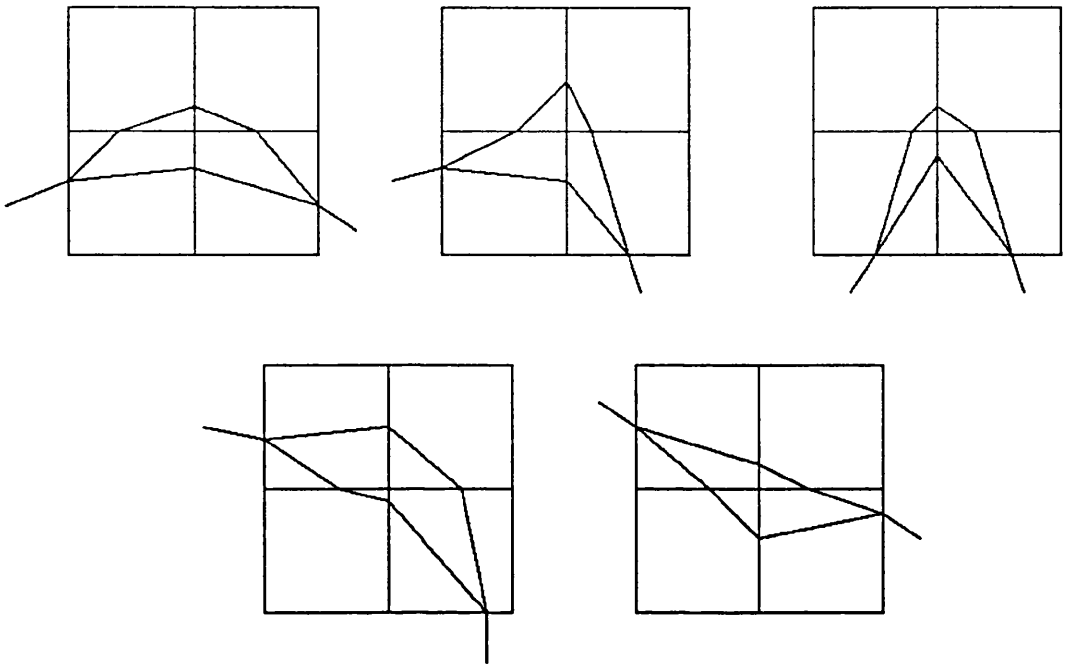


Figure 5.27

The contribution of the total penalty from all 4 pixels is calculated for each of the two routes, with the line segments chosen optimally for that route. The points at which the line segments enter and leave the 4 pixels are fixed and convergence to a local minimum is reached within a few iterations. Although we are considering only small changes in the route, changes to routes further afield may take place as a result of several small changes.

Returning to the example, we see in Figure 5.28 the reconstruction obtained when the route changing development is included in the algorithm. Because the route is only changed when it leads to a reduction in the penalty, reconstructions obtained in this way have penalties no greater than those obtained without route changing.

The updating process in the above line fitting procedure has the general characteristics of an ICM method: the penalty is minimised with respect to one component of the boundary whilst everything else is held fixed. This method will

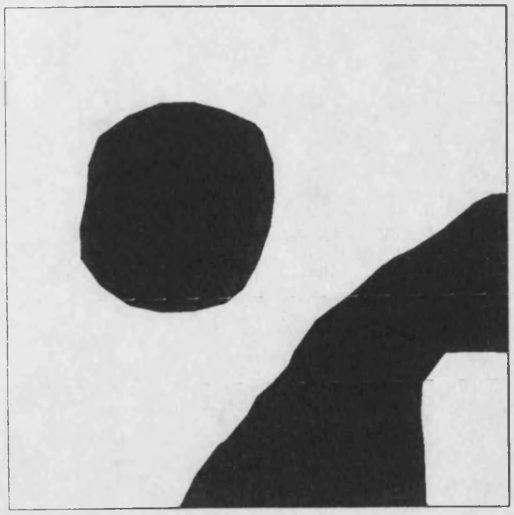


Figure 5.28

generally yield a local minimum of the penalty and it is possible that the final reconstruction could be improved further by making a number of route changes simultaneously. For example, the penalty might be reduced by moving a long vertical edge one pixel to the left whereas it would increase initially if only one route change were made at a time.

To allow further exploration of alternative routes we have implemented a form of simulated annealing. This method retains the property that for a given route the point on a pixel edge at which two line segments meet is chosen optimally. However, when comparing the minimum penalties for *different* routes we allow the route with the larger penalty to be chosen with non-zero probability. Suppose two routes, A and B, have minimum penalties pen_A and pen_B, then, when the annealing process is at temperature T we select route A and its optimal edges with probability

$$\frac{e^{(-pen_A/T)}}{e^{(-pen_A/T)}+e^{(-pen_B/T)}}$$

otherwise we choose route B. Of course, only the contribution to the total penalty from the four pixels concerned need actually be calculated.

By restricting the random choice to the route alone, we ensure that, effectively, the annealing process is applied to a fairly low dimension problem, the number of variables being of the order of the number of boundary pixels.

Theorem B of Geman and Geman (1984) demonstrates the convergence of their simulated annealing method. In its stated form, this theorem does not apply to our hybrid procedure whose iterative steps combine a random choice of route with a deterministic choice of line segments given that route and currently fixed end points. Perhaps a sufficiently general result could be proved but this would, presumably, still only apply for gentle cooling schedules. However, we prefer to think of the annealing method simply as a convenient numerical procedure with which to search a little further afield than the ICM approach.

The best results were obtained using a cooling schedule in which T decreased logarithmically from 3.5 to 0.5 over several hundred sweeps and linearly from 0.5 to zero over several hundred more. We then continued to update using T=0 until convergence, which usually required only a few sweeps. This corresponds to ICM and guarantees convergence to a local minimum. Although simulated annealing often produced a lower penalty, we have found that the computational extravagance of the method is not justified by the very small difference in the reconstructions compared with those obtained using the local minimisation procedure.

5.5.7 Some more examples

Using the true scene shown in Figure 5.23 we experimented with different levels of noise and values of the parameter γ. The results are most interesting and provide an insight into the mechanics of the program, particularly with regard to the route changing. Where the value of γ is small the process attempts to find the line segments which give a proportion of black very close to the record for each pixel. This is illustrated by the reconstruction in Figure 5.29 which was obtained using the version of the program which did not include the route changing development with γ=0.1. When the route changing is included the reconstruction in Figure 5.30 is obtained. This is an amazing reconstruction which demonstrates that by considering only small changes in the route it is possible to find routes which are much further afield. When larger values of γ are used, more emphasis is placed on reducing the length of the edge in the reconstruction. Figures 5.31 and 5.32 show the reconstructions obtained using γ=20 without and with the route changing respectively. Although objects may become very small, they are constrained to remain as the routes surrounding them may not disappear

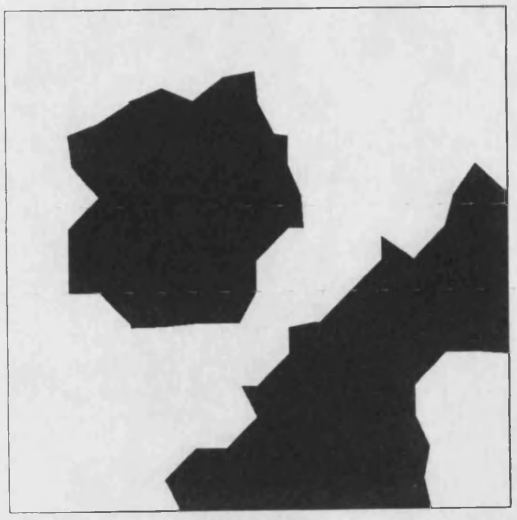


Figure 5.29
Reconstruction technique: Line fitting
without route changing
Parameter: γ=0.1
Initial routes obtained from Figure 5.24

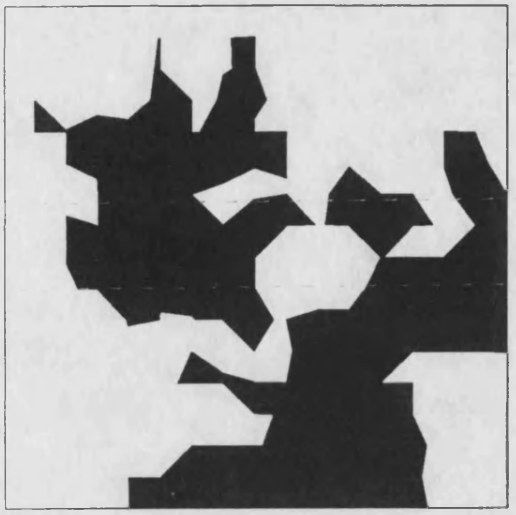


Figure 5.30
Reconstruction technique: Line fitting
with route changing
Parameter: γ=0.1
Initial routes obtained from Figure 5.24

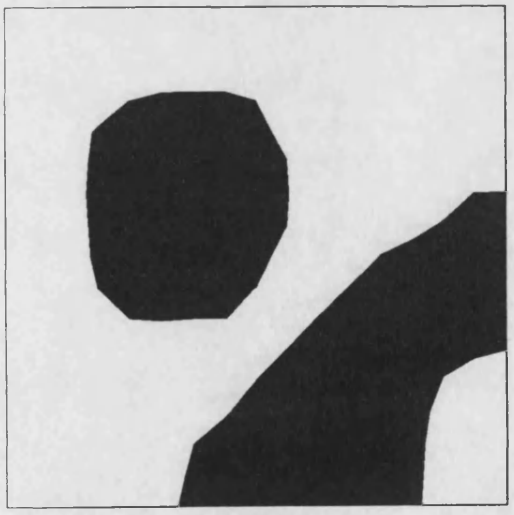


Figure 5.31
Reconstruction technique: Line fitting
without route changing
Parameter: γ=20
Initial routes obtained from Figure 5.24

Figure 5.32
Reconstruction technique: Line fitting
with route changing
Parameter: γ=20
Initial routes obtained from Figure 5.24

completely. At larger noise levels the reliability of full and quarter pixel reconstructions becomes suspect and they may lead to unsatisfactory routes for the line fitting stage. This is illustrated by the reconstruction in Figure 5.33. Because the quarter pixel reconstruction contains only one area of black just one route is found and the line segment reconstruction consequently suffers. Although it would add to the computational complexity, it is possible that the program could be further developed so that routes could be introduced and removed during the process. In general, where the variance is low, the quarter pixel reconstruction provides a good estimate and any routes which are found are fairly reliable starting points. Figure 5.34 shows the reconstruction obtained using γ=4 when σ^2=0.01. Figure 5.35 shows the reconstruction obtained using γ=200 when σ^2=0.0001 which demonstrates the ability of the method to reconstruct this image almost perfectly where the noise is very low.

In testing and developing the program we have experimented with many different examples. The best results are obtained from those scenes which do not contain jagged points or irregularities. Also when the pixel grid is very coarse the quality of the reconstruction can depend quite heavily on the positioning of the colourings relative to the pixel grid, particularly if it leads to a pixel containing more than 2 areas of colour.

5.5.8 Programming the line fitting

The programming for the line fitting process was done in several stages. The first of these was to decide the general approach and data storage techniques that would be used. In this section we give a brief outline of this stage. The next stage was to convert the quarter pixel reconstruction into a route form, this proved more complex than was originally envisaged and involved making provision for many special cases. The final stage involved programming the actual updating, including the route changing and simulated annealing. Again this proved a painstaking procedure involving the consideration of many problems caused by special cases.

The eventual output of the program is a file which contains the coordinates of the region, the number of routes within the region and the coordinates of the joining points of the line segments. This file is then used to reproduce the reconstruction either on a computer screen or as a hard copy. The line fitting

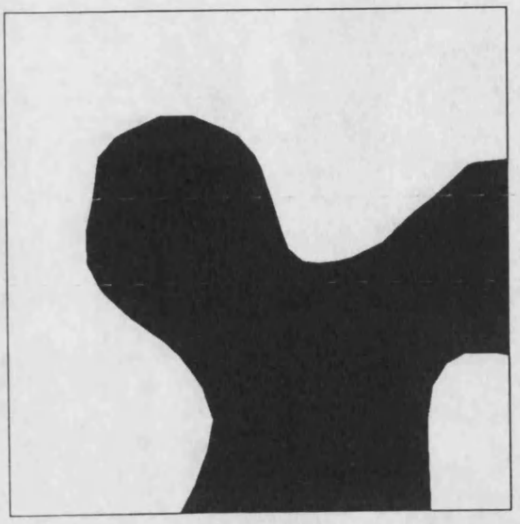


Figure 5.33
Reconstruction technique: Line fitting
with route changing
Parameters: $\sigma^2=0.5$
$\gamma=4$

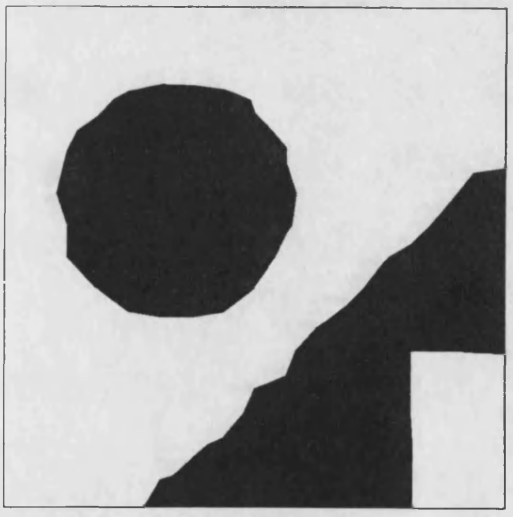


Figure 5.34
Reconstruction technique: Line fitting
with route changing
Parameters: $\sigma^2=0.01$
$\gamma=4$

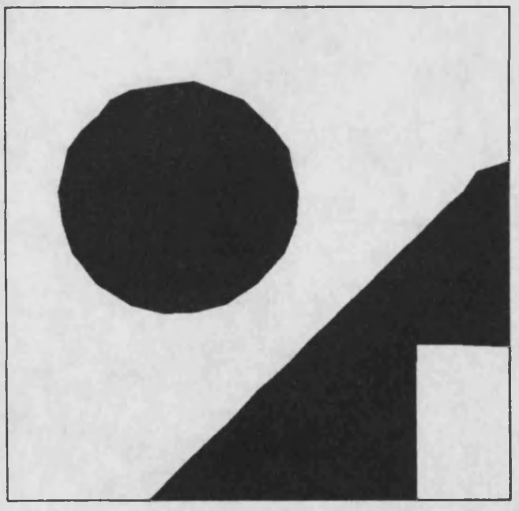


Figure 5.35
Reconstruction technique: Line fitting
with route changing
Parameters: $\sigma^2=0.0001$
$\gamma=200$

program was written in **FORTRAN** as was the postscript conversion filter which produces the hard copy pictures. The computer display program was written in **C** and uses the **SUN CGI** graphics package.

The general approach and data storage techniques

As we have already stated, the image is regarded as a series of line segments dividing white and black areas of colour. The routes are found and we choose to update the pixels around each route until convergence. An alternative technique would be to store the information locally for each pixel and to update pixels in a raster fashion. Such a technique has the advantage of avoiding much of the complexities of setting up and storing the routes. However, the main advantage of the route tracing approach that we have described is that when route changing is used, the directional changes of the route are already available and do not need to be recalculated at every updating.

Three structures provide the basis of the storage used in the program:

EdgeInformation(*MAXNUMEDGES*, 3)

StartofRoutes(*MAXNUMROUTES*, 4)

and **AlreadyUsed**(*NumberofRows, NumberofColumns*)

where *MAXNUMEDGES* is the maximum number of line segments which are allowed and *MAXNUMROUTES* is the maximum number of different routes which are allowed.

For each edge, 3 pieces of information are held in **EdgeInformation:**

(i) The relative direction (0-3) to the pixel on the route which will be visited next.

(ii) The distance along the join (of this pixel and the next on the route) at which the edge cuts, measured from the black side. This value is referred to as the cut value.

(iii) The pointer to the next cell in **EdgeInformation** which holds information relating to the next pixel on the route.

Before the route changing was incorporated (iii) was not required since each pixel led to the next cell in **EdgeInformation.**

For each route, 4 pieces of information are held in **StartofRoutes:**

(i) The horizontal coordinate of the first pixel on the route.

(ii) The vertical coordinate of the first pixel on the route.

(iii) The direction (0-3) that the route moves to the 2nd pixel on the route.

(iv) A pointer to the position in **EdgeInformation** that holds the information for the 2nd pixel on the route.

When the route changing is incorporated this array may be updated, as starting pixels will sometimes be eliminated from the route.

AlreadyUsed is a flag array which indicates whether or not a pixel has already been used. This is required because each pixel may only lie on one route in a reconstruction and then only once on that route.

5.5.9 Extending the line fitting method to *k* colour scenes

So far we have demonstrated the effectiveness of the line fitting method in reconstructing 2 colour images. In this section we propose an extension of the technique which handles images containing more than 2 colours, and use a simple example to illustrate how the technique may be used.

In any full pixel reconstruction the region contains areas which are all one colour, areas which contain boundaries between two colours and areas where three or more colours meet. If the areas which contain two colours split by a boundary are identified then the line fitting method may be applied to these areas, one by one.

We illustrate this possibility by a simple example. Figure 5.36 shows the true scene which contains 3 colours, each pixel taking value 0,1 or 2. A 16 by 16 grid is superimposed on the image and for each pixel the average colouring is recorded. (This is equivalent to using the proportion of black for 2 colour scenes.) Gaussian noise with mean 0 and variance 0.05 is added to the true value for each pixel, giving the record. Figure 5.37 shows the ICM reconstruction obtained using $\beta_1=1.0$ and $\beta_2=\beta_1/\sqrt{2}$. The prior model which is used for the ICM penalises all discrepant pairs equally, although this may not be the "best" prior for this particular case the level of noise is small and it is unlikely that using other more intricate priors would give superior results. Inspection of this reconstruction reveals that the region may be partitioned into areas containing just one colour and areas which contain boundaries between two colours. By

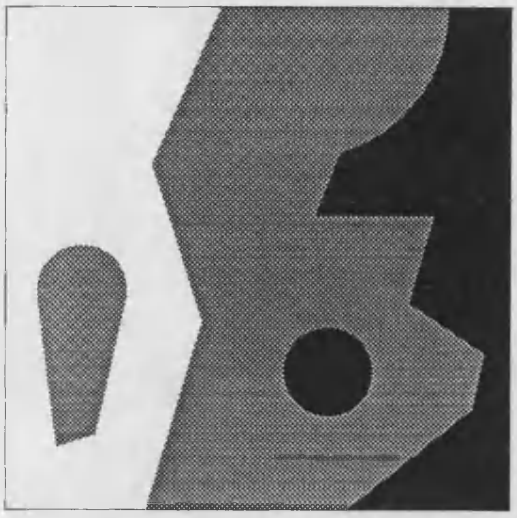


Figure 5.36
The true scene

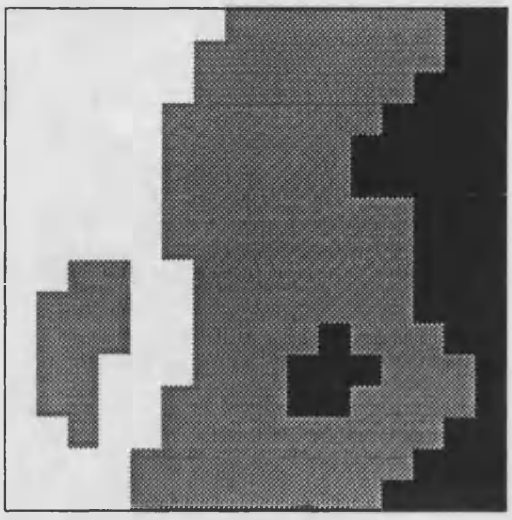


Figure 5.37
Reconstruction technique: ICM
Parameters:

$\beta_1 = 1.0$

$\beta_2 = \beta_1/\sqrt{2}$

Initial estimate: Closest mean classifier

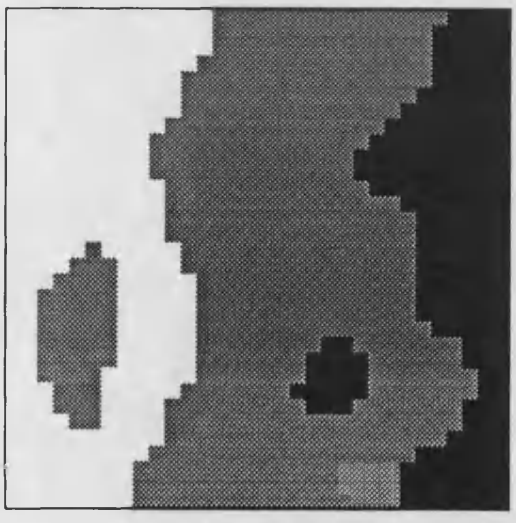


Figure 5.38
Reconstruction technique: Jennison's quarter pixel technique
Parameters: $\beta_1 = 1.0$

$\beta_2 = \beta_1/\sqrt{2}$

Initial estimate: ICM reconstruction

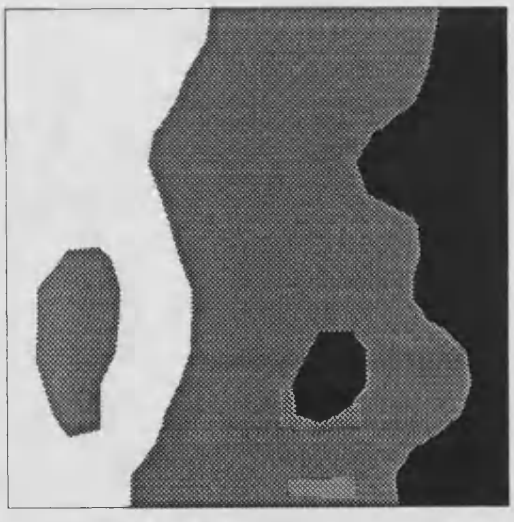


Figure 5.39
Reconstruction technique: Line fitting with route changing
Parameter: $\gamma = 4.0$
Initial routes obtained from Figure 5.38

considering the areas which contain two colours we may find routes which will be used for the line fitting method. As an intermediate step, quarter pixel reconstructions are found for each of the areas. These are shown (in a combined form) in Figure 5.38. The two colour line fitting method is then applied to each of the routes with $\gamma=4$, giving the reconstruction shown in Figure 5.39.

We have treated this example in a special way, avoiding the complexities of identifying the different areas of the region. The technique has not been implemented, a modification of the 2 colour program was used to obtain the reconstruction. However the success of the technique in reconstructing the image demonstrates the potential of the method to handle more intricate cases.

5.5.10 The equations for different line segment configurations

As we described in Section 5.5.3, updating the line segments involves using an iterative procedure to obtain the point on the common pixel edge at which the line segments should meet. In this section we show the equations which are used for each of the six different cases.

Case 1.

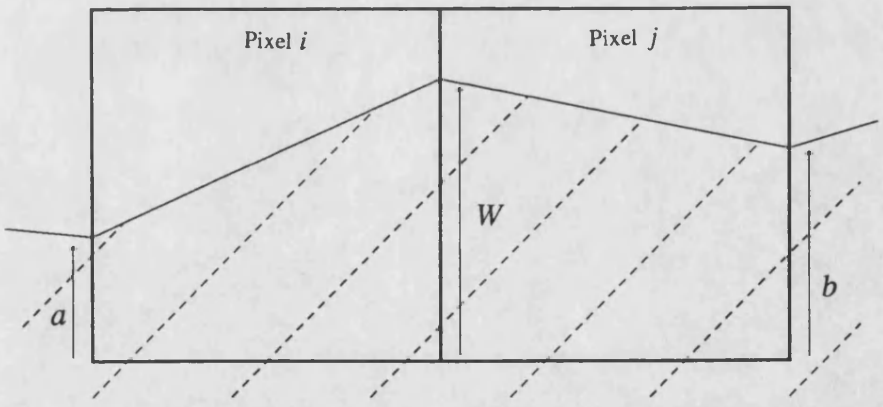


Figure 5.40

The contribution to the objective function for the case in Figure 5.40 is given by

$$g(W) = \frac{1}{2\sigma^2}\left[(y_i-a-\frac{(W-a)}{2})^2+(y_j-b-\frac{(W-b)}{2})^2\right] + \gamma\{\sqrt{1+(W-a)^2}+\sqrt{1+(W-b)^2}\}.$$

Differentiating we obtain

$$\frac{dg(W)}{dW} = \frac{1}{4\sigma^2}(2W+a-2y_i+b-2y_j) + \gamma\left[\frac{(W-a)}{\sqrt{1+(W-a)^2}}+\frac{(W-b)}{\sqrt{1+(W-b)^2}}\right].$$

Given an approximate solution W_{s-1} we solve

$$\frac{1}{4\sigma^2}(2W_s+a-2y_i+b-2y_j) + \gamma\left[\frac{(W_s-a)}{\sqrt{1+(W_{s-1}-a)^2}}+\frac{(W_s-b)}{\sqrt{1+(W_{s-1}-b)^2}}\right] = 0$$

to obtain

$$W_s = \frac{4\sigma^2\gamma\left[\dfrac{a}{\sqrt{1+(W_{s-1}-a)^2}}+\dfrac{b}{\sqrt{1+(W_{s-1}-b)^2}}\right] + (2y_i-a+2y_j-b)}{2+4\sigma^2\gamma\left[\dfrac{1}{\sqrt{1+(W_{s-1}-a)^2}}+\dfrac{1}{\sqrt{1+(W_{s-1}-b)^2}}\right]}.$$

Case 2.

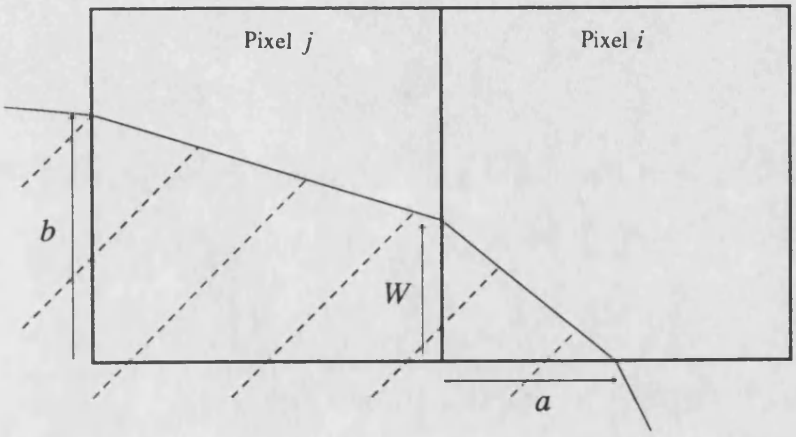


Figure 5.41

The contribution to the objective function for the case in Figure 5.41 is given by

$$g(W) = \frac{1}{2\sigma^2}\left[(y_i-\frac{Wa}{2})^2+(y_j-b-\frac{(W-b)}{2})^2\right] + \gamma\{\sqrt{W^2+a^2}+\sqrt{1+(W-b)^2}\}.$$

Differentiating we obtain

$$\frac{dg(W)}{dW} = \frac{1}{4\sigma^2}(Wa^2-2ay_i+W+b-2y_j) + \gamma\left[\frac{W}{\sqrt{W^2+a^2}}+\frac{(W-b)}{\sqrt{1+(W-b)^2}}\right].$$

Given an approximate solution W_{s-1} we solve

$$\frac{1}{4\sigma^2}(W_sa^2-2ay_i+W_s+b-2y_j) + \gamma\left[\frac{W_s}{\sqrt{W_{s-1}^2+a^2}}+\frac{(W_s-b)}{\sqrt{1+(W_{s-1}-b)^2}}\right] = 0$$

to obtain

$$W_s = \frac{4\sigma^2\gamma\left[\dfrac{b}{\sqrt{1+(W_{s-1}-b)^2}}\right] + (2ay_i+2y_j-b)}{(1+a^2)+4\sigma^2\gamma\left[\dfrac{1}{\sqrt{W_{s-1}^2+a^2}}+\dfrac{1}{\sqrt{1+(W_{s-1}-b)^2}}\right]}.$$

Case 3.

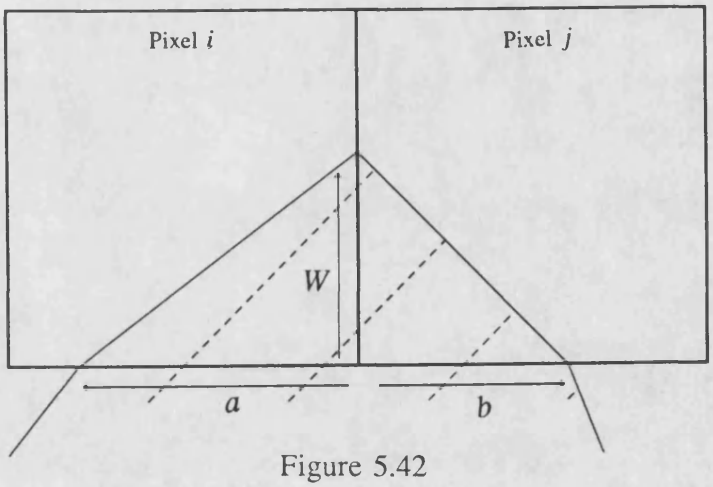


Figure 5.42

The contribution to the objective function for the case in Figure 5.42 is given by

$$g(W) = \frac{1}{2\sigma^2}\left[\left(y_i-\frac{Wa}{2}\right)^2+\left(y_j-\frac{Wb}{2}\right)^2\right\} + \gamma\{\sqrt{W^2+a^2}+\sqrt{W^2+b^2}\right].$$

Differentiating we obtain

$$\frac{dg(W)}{dW} = \frac{1}{4\sigma^2}(Wa^2-2ay_i+Wb^2-2by_j) + \gamma\left[\frac{W}{\sqrt{W^2+a^2}}+\frac{W}{\sqrt{W^2+b^2}}\right].$$

Given an approximate solution W_{s-1} we solve

$$\frac{1}{4\sigma^2}(W_sa^2-2ay_i+W_sb^2-2by_j) + \gamma\left[\frac{W_s}{\sqrt{W_{s-1}^2+a^2}}+\frac{W_s}{\sqrt{W_{s-1}^2+b^2}}\right] = 0$$

to obtain

$$W_s = \frac{2ay_i+2by_j}{(a^2+b^2)+4\sigma^2\gamma\left[\dfrac{1}{\sqrt{W_{s-1}^2+a^2}}+\dfrac{1}{\sqrt{W_{s-1}^2+b^2}}\right]}$$

Case 4.

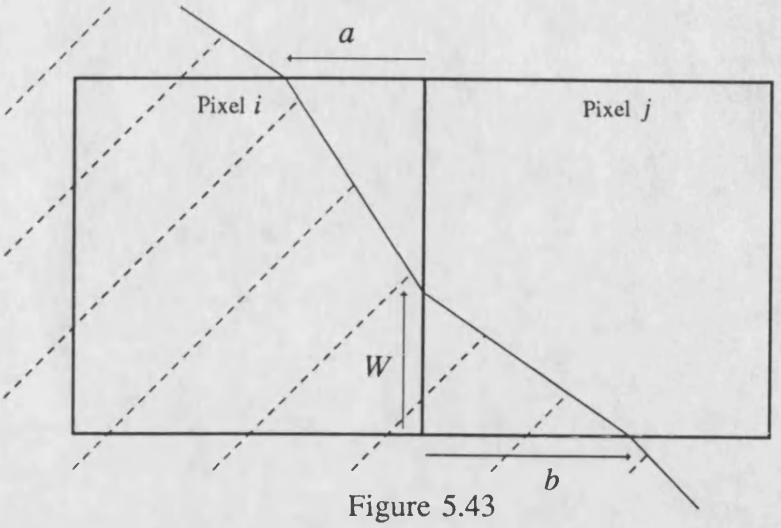


Figure 5.43

The contribution to the objective function for the case in Figure 5.43 is given by

$$g(W) = \frac{1}{2\sigma^2}\left[y_i-(1-\frac{(1-W)a}{2})\right]^2+(y_j-\frac{Wb}{2})^2 + \gamma\{\sqrt{(1-W)^2+a^2}+\sqrt{W^2+b^2}\}.$$

Differentiating we obtain

$$\frac{dg(W)}{dW} = \frac{1}{4\sigma^2}\left[Wa^2-2a(1-\frac{a}{2}-y_i)+Wb^2-2by_j\right] + \gamma\left[\frac{W-1}{\sqrt{(1-W)^2+a^2}}+\frac{W}{\sqrt{W^2+b^2}}\right].$$

Given an approximate solution W_{s-1} we solve

$$\frac{1}{4\sigma^2}\left[W_sa^2-2a(1-\frac{a}{2}-y_i)+W_sb^2-2by_j\right] + \gamma\left[\frac{W_s-1}{\sqrt{(1-W_{s-1})^2+a^2}}+\frac{W_s}{\sqrt{W_{s-1}+b^2}}\right] = 0$$

to obtain

$$W_s = \frac{2by_j-2a(1-\frac{a}{2}-y_i)+\frac{4\sigma^2\gamma}{\sqrt{(1-W_{s-1})^2+a^2}}}{(a^2+b^2)+4\sigma^2\gamma\left[\frac{1}{\sqrt{(1-W_{s-1})^2+a^2}}+\frac{1}{\sqrt{W_{s-1}^2+b^2}}\right]}.$$

Case 5

The contribution to the objective function for the case in Figure 5.44 is given by

$$g(W) = \frac{1}{2\sigma^2}\left[y_i-a-\frac{(W-a)}{2}\right]^2 + \gamma\sqrt{1+(W-a)^2}.$$

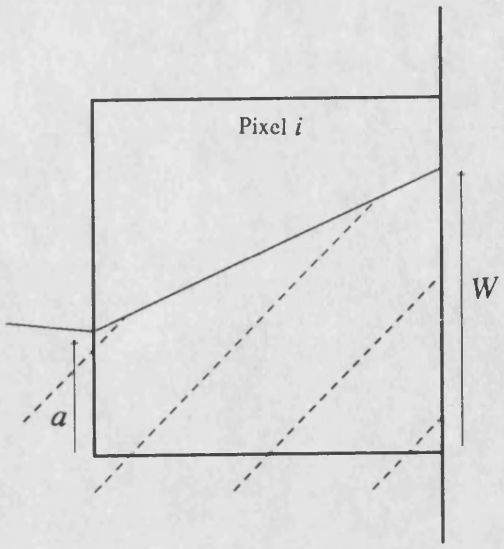


Figure 5.44

Differentiating we obtain

$$\frac{dg(W)}{dW} = \frac{1}{4\sigma^2}(W+a-2y_i) + \frac{\gamma(W-a)}{\sqrt{1+(W-a)^2}}.$$

Given an approximate solution W_{s-1} we solve

$$\frac{1}{4\sigma^2}(W_s+a-2y_i) + \frac{\gamma(W_s-a)}{\sqrt{1+(W_{s-1}-a)^2}}=0$$

to obtain

$$W_s = \frac{\dfrac{4\sigma^2\gamma a}{\sqrt{1+(W_{s-1}-a)^2}}+(2y_i-a)}{1+\left[\dfrac{4\sigma^2\gamma}{\sqrt{1+(W_{s-1}-a)^2}}\right]}.$$

Case 6

The contribution to the objective function for the case in Figure 5.45 is given by

$$g(W) = \frac{1}{2\sigma^2}\left[\frac{Wa}{2}-y_i\right]^2 + \gamma\sqrt{W^2+a^2}.$$

Differentiating we obtain

$$\frac{dg(W)}{dW} = \frac{1}{4\sigma^2}(Wa^2-2ay_i) + \frac{\gamma W}{\sqrt{W^2+a^2}}.$$

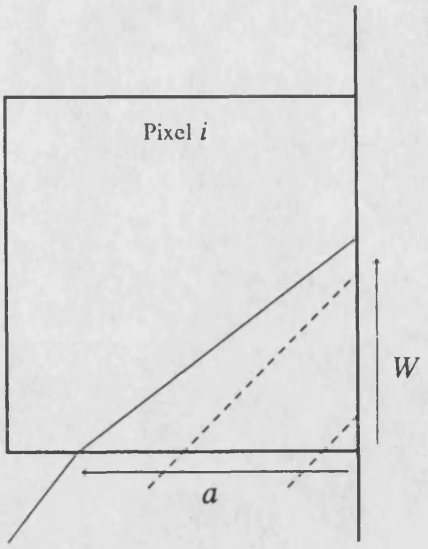


Figure 5.45

Given an approximate solution W_{s-1} we solve

$$\frac{1}{4\sigma^2}(W_s a^2 - 2ay_i) + \frac{\gamma W_s}{\sqrt{W_{s-1}^2 + a^2}}.$$

to obtain

$$W_s = \frac{2ay_i}{a^2 + \dfrac{1}{\sqrt{W_{s-1}^2 + a^2}}}.$$

5.6 Combining the cascade algorithm with the line fitting method

When the variance of the record is high there is often little point in using any form of subpixel refinement. However, as we have demonstrated in Section 4, the variance of the record may be reduced using aggregation and the quality of reconstructions improved. In this section we consider an example in which the noise level is high and show how the cascade algorithm may be successfully used in conjunction with the line fitting method to produce high quality reconstructions.

The example

Figure 5.46 shows the true scene. A 256 by 256 pixel grid was superimposed on this scene and the record was obtained by adding Gaussian noise with variance 4 to the proportion of black in each pixel. Figure 5.47 shows the closest mean classifier for this record. Progressing through the cascade algorithm produces the

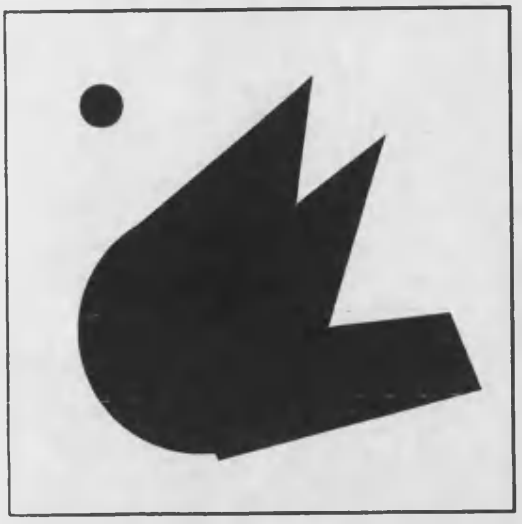


Figure 5.46

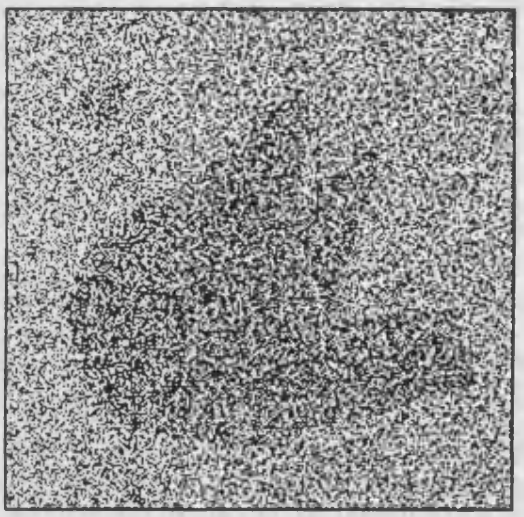


Figure 5.47

reconstructions shown in Figures 5.48 to 5.53; reconstructions obtained on the coarsest grids are not shown. As we have found in previous examples, it is not necessary to complete the cascade as satisfactory reconstructions may be obtained at coarser levels. The 32 by 32 record has variance 1/16 and we can use this record to obtain a subpixel reconstruction. Using the 32 by 32 cascade reconstruction as an initial estimate, Jennison's quarter pixel technique is applied and the resulting reconstruction is used as a starting point for the line fitting method. This 3-step process may be applied at any grid level although the best results are obtained when the variance is low and the cascade reconstruction has retained many of the features of the image. Figures 5.54 and 5.55 show the line segment reconstructions for the 32 by 32 and 64 by 64 records respectively. Given the level of noise in the original record and the irregular nature of parts of the true scene, these reconstructions are very good and demonstrate both the power of the proposed method and its limitations.

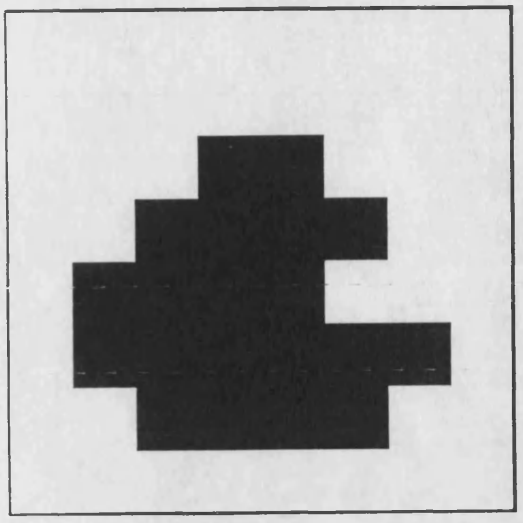


Figure 5.48
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 4 × 4 cascade
Dimensions: 8 × 8
Record: 256 × 256 aggregated 5 times

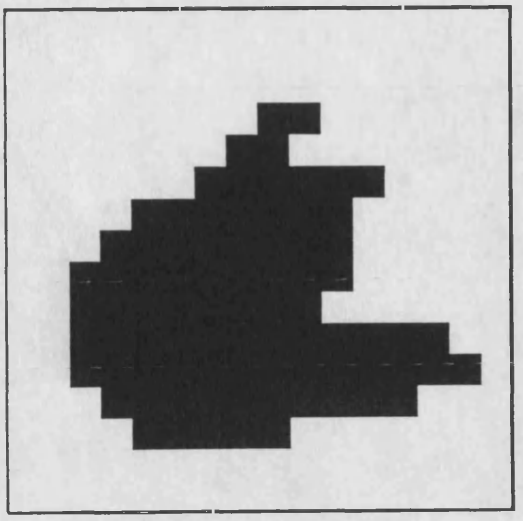


Figure 5.49
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 8 × 8 cascade (Fig 5.48)
Dimensions: 16 × 16
Record: 256 × 256 aggregated 4 times

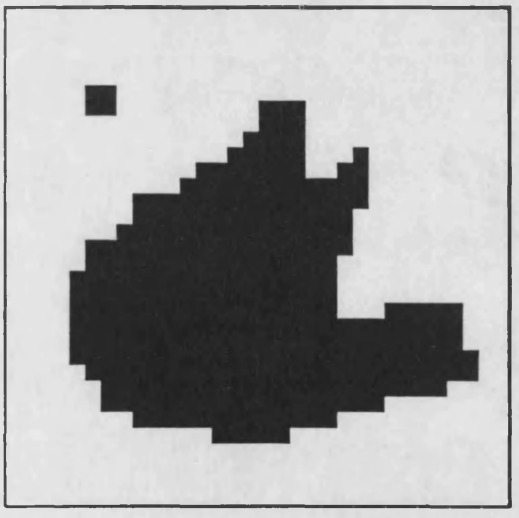


Figure 5.50
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 16 × 16 cascade (Fig 5.49)
Dimensions: 32 × 32
Record: 256 × 256 aggregated 3 times

Figure 5.51
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 32 × 32 cascade (Fig 5.50)
Dimensions: 64 × 64
Record: 256 × 256 aggregated twice

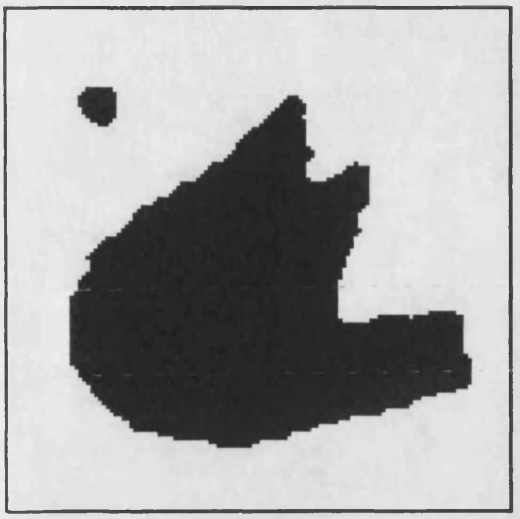


Figure 5.52
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 64 × 64 cascade (Fig 5.51)
Dimensions: 128 × 128
Record: 256 × 256 aggregated

Figure 5.53
Reconstruction technique: Cascade, ICM
Parameters: $\beta_1 = 1.0$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: 128 × 128 cascade (Fig 5.52)
Dimensions: 256 × 256
Record: 256 × 256 (original)

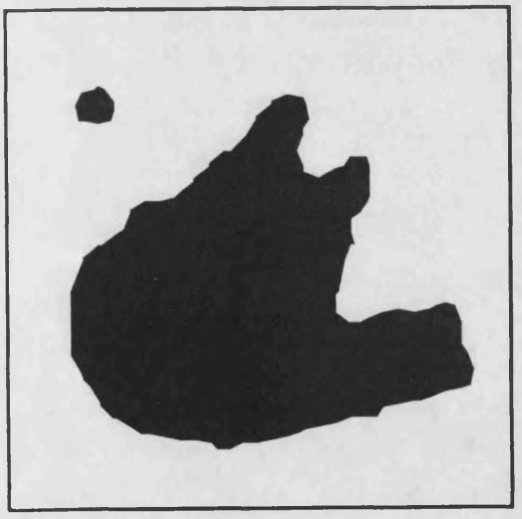


Figure 5.54
Reconstruction technique: Line fitting with route changing, via cascade
Grid size: 32 by 32
Parameter: γ=4

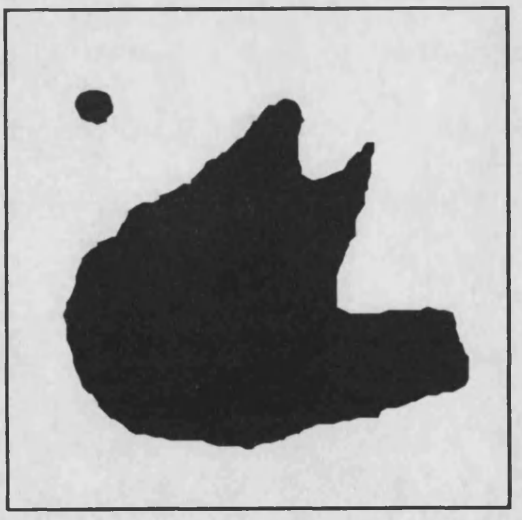


Figure 5.55
Reconstruction technique: Line fitting with route changing, via cascade
Grid size: 64 by 64
Parameter: γ=4

Chapter 6: A test for detecting oversmoothing

6.1 Introduction

When some reconstruction techniques are used, small changes in the value of β can have large effects on the reconstruction obtained. This is particularly true of exact MAP reconstruction, described in Section 3.4, where areas of colour can be obliterated as a result of oversmoothing. For any reconstruction we may calculate the difference between the fitted value x_i at pixel i and its record y_i. This value is known as the residual at pixel i. If an area of colour has been obliterated as a result of oversmoothing then the residuals at those pixels which have been misclassified are expected to be larger than those occurring at pixels which have been correctly classified. We present a graphical aid which assists in the subjective analysis of the grouping of the residuals, and propose a statistical test which aims to detect any significant groupings. The idea of the test is that by inspection of the spatial positioning of the largest of the residuals for a given reconstruction, we may establish whether there are any grounds to suspect that the image has been oversmoothed. We deal specifically with the two colour case, where each pixel x_i takes value 0 or 1, representing white and black respectively.

The residuals

We define a residual r_i as being

$$r_i = \begin{cases} y_i - x_i & \text{if } x_i = 0 \\ -(y_i - x_i) & \text{if } x_i = 1 \end{cases}$$

or equivalently

$$r_i = (y_i - x_i)(1 - 2x_i)$$

The residuals are calculated in this way so that the colouring of any pixel which has a residual greater than 0.5 has been affected by the smoothing that has been employed, and were β to be reduced sufficiently the colouring of the pixel would change to that of its closest mean. For example, consider two pixels i and j which are classified as black in the reconstruction (i.e. $x_i=1$ and $x_j=1$) and have records $y_i=-1.0$ and $y_j=3.0$. Although both may be correctly classified, the colouring of pixel i has been affected by level of smoothing whereas the colouring of pixel j would not change were a smaller value of β used.

We are particularly interested in those pixels with large residuals. In looking at the significance of a residual it is natural to consider the variance of the record and to use the likelihood ratio as a measure of closeness of fit. The likelihood ratio for the additive Gaussian case is given by

$$\frac{f(y_i \mid x_i^* = 1 - x_i)}{f(y_i \mid x_i^* = x_i)} = \exp\left\{-\frac{1}{2\sigma^2}[(y_i - (1-x_i))^2 - (y_i - x_i)^2]\right\}$$

$$= \exp\left\{-\frac{1}{2\sigma^2}(2y_i - 1)(2x_i - 1)\right\}$$

Pixel i is then defined to house an *informative residual* if the log-likelihood ratio is greater than some specified value k. i.e.

$$-\frac{1}{2\sigma^2}(2y_i - 1)(2x_i - 1) > k.$$

An alternative and equivalent definition is that r_i is informative if $r_i > 0.5 + k\sigma^2$.

Oversmoothing

Figure 6.1 shows a true scene in which each pixel is coloured wholly one colour.

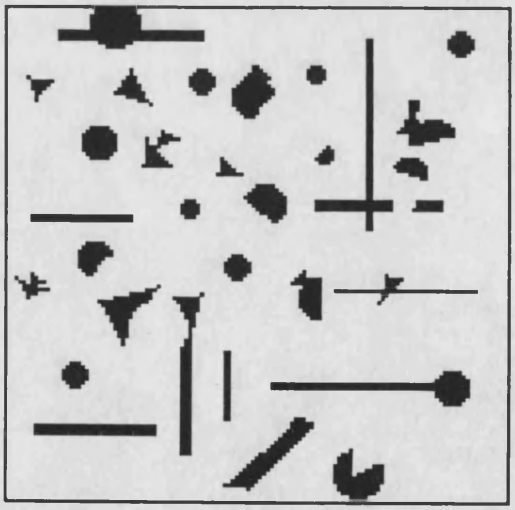


Figure 6.1

Gaussian noise with mean zero and variance 0.3 is added to the true value for each pixel and a MAP reconstruction is obtained using $\beta_1 = 0.8$ and $\beta_2 = \beta_1/\sqrt{2}$; this is shown in Figure 6.2. Several small areas of colour have been misclassified in this reconstruction. The residuals for this reconstruction are calculated and

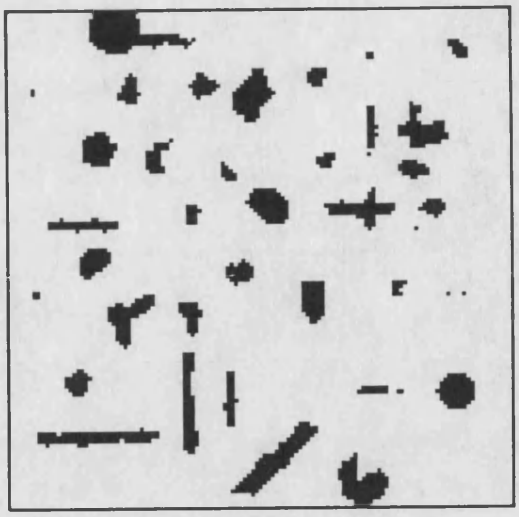


Figure 6.2

Figure 6.3 highlights the informative residuals when k=2. From this plot it is fairly clear that the informative residuals are grouped together in certain parts of the region. It is more helpful to look at the residuals alongside the reconstruction and the two may be combined in the form shown in Figure 6.4, where areas of black in the reconstruction have been outlined. The number of informative residuals which appear depends on the value of k that is used. In this example there are 390 informative residuals for k=2. Figures 6.5-6.8 show the plots for k= 0, 1, 3 and 4 respectively. This method of displaying the residuals provides an excellent graphical aid for diagnosing oversmoothing. However, although we may suspect that the record has been oversmoothed, we have no statistical evidence to justify this. We now propose a test which uses a Monte Carlo simulation technique to establish the significance of the grouping.

6.2 The test

We set up the following hypothesis:

H_0: The reconstruction is essentially correct, certainly away from the edges of areas of colour in the reconstruction.

H_1: The record has been oversmoothed, resulting in the obliteration of areas of colour.

When the null hypothesis is true the raw residual at each pixel is nearly independent of neighbouring residuals. This is because the colouring x_i at pixel i will, in general, be equal to the true colouring x_i^* at pixel i. This implies that

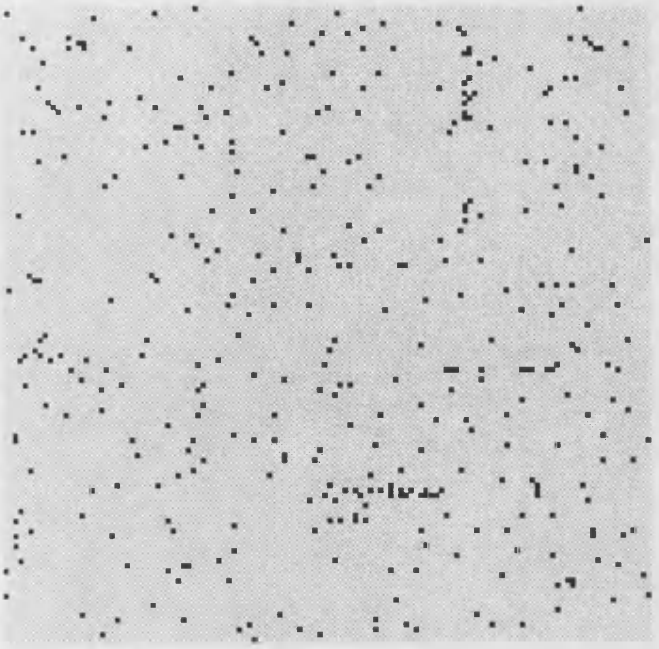


Figure 6.3
Reconstruction Parameters:
Technique: exact MAP
$\beta_1 = 0.8$
$\beta_2 = \beta_1/\sqrt{2}$

Residual Parameters:
$k=2$
390 informative residuals

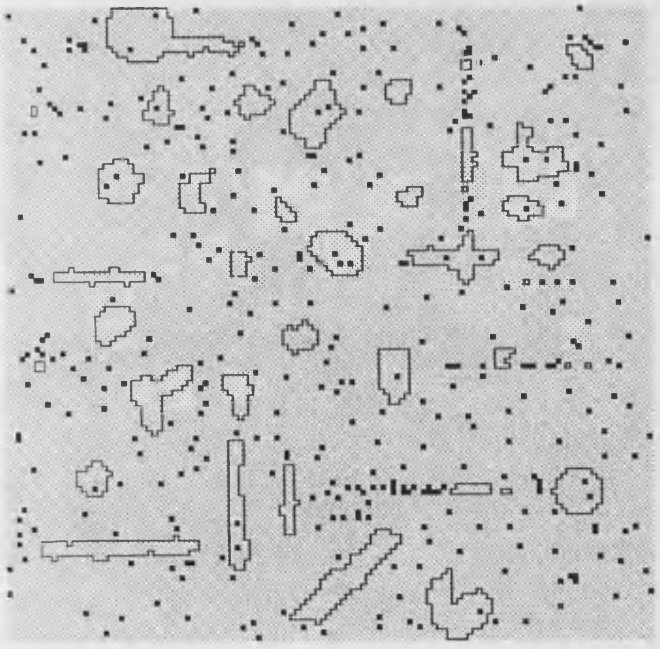


Figure 6.4
Reconstruction Parameters:
Technique: exact MAP
$\beta_1 = 0.8$
$\beta_2 = \beta_1/\sqrt{2}$

Residual Parameters:
$k=2$
390 informative residuals

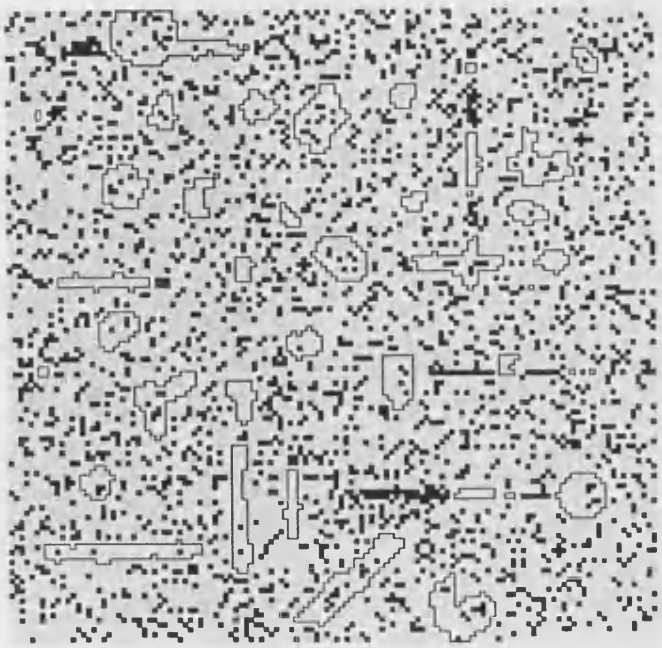


Figure 6.5
Reconstruction Parameters:
Technique: exact MAP
$\beta_1 = 0.8$
$\beta_2 = \beta_1/\sqrt{2}$
Residual Parameters:
$k = 0$
2652 informative residuals

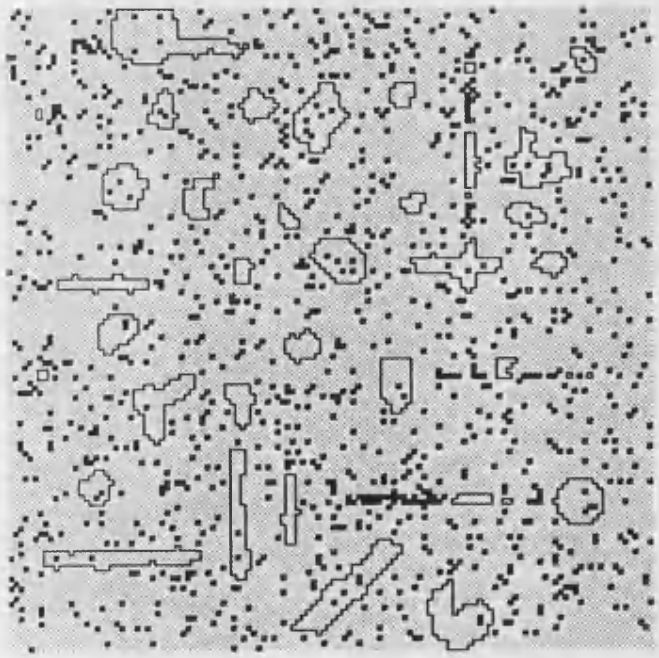


Figure 6.6
Reconstruction Parameters:
Technique: exact MAP
$\beta_1 = 0.8$
$\beta_2 = \beta_1/\sqrt{2}$
Residual Parameters:
$k = 1$
1151 informative residuals

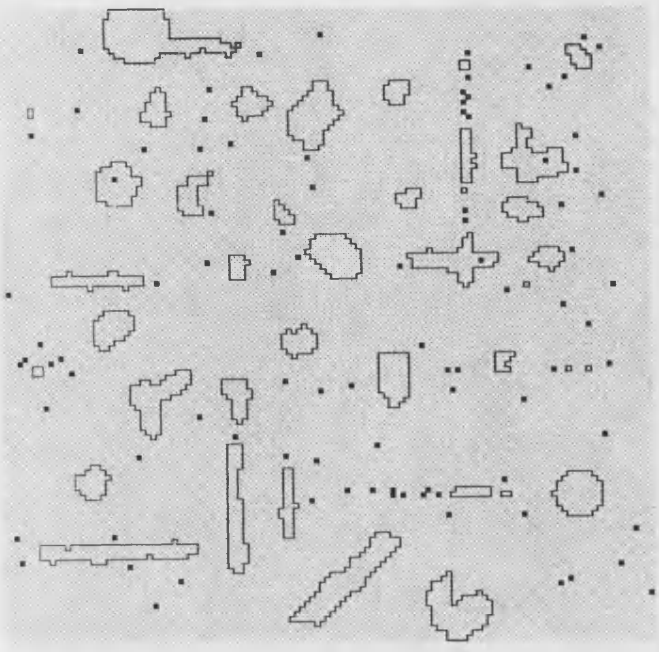


Figure 6.7
Reconstruction Parameters:
Technique: exact MAP
$\beta_1 = 0.8$
$\beta_2 = \beta_1/\sqrt{2}$

Residual Parameters:
$k=3$
93 informative residuals

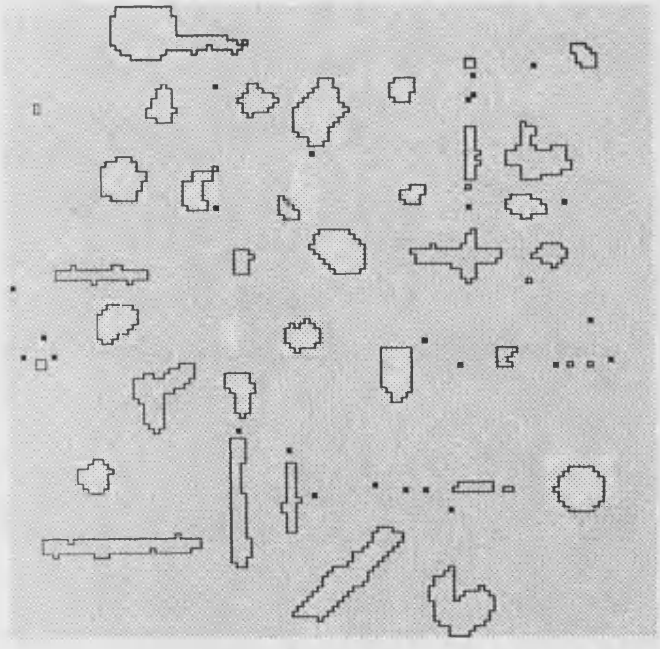


Figure 6.8
Reconstruction Parameters:
Technique: exact MAP
$\beta_1 = 0.8$
$\beta_2 = \beta_1/\sqrt{2}$

Residual Parameters:
$k=4$
25 informative residuals

$y_i - x_i$ will usually be equal to $y_i - x_i^*$ which is independently distributed as Gaussian with mean zero and variance σ^2. Thus the occurrence of informative residuals within the accepted region (which excludes pixels on the edges of areas of colour) is uniformly random. The test is based on the size of the largest grouping of informative residuals occurring away from the edges of areas of colour. We define a grouping as being a set of informative residuals which are connected to each other, either directly or via other informative residuals in the set. The connection may be to horizontally, vertically or diagonally adjacent pixel sites although the connecting criteria may be chosen to correspond to the neighbourhood system that is used in reconstruction. For example, if a first order model (in which only horizontally and vertically adjacent pixels are regarded as neighbours) was used in reconstruction, then only groupings in which the connections are horizontal or vertical are considered.

Informative residuals which occur on the edges of areas of colour in the reconstruction are not included in any analysis, nor are they shown in graphical plots. This is because the residuals at edges offer little information about oversmoothing. Inherent in the choice of the value of β that is used is the acceptance that certain edge configurations will be smoothed. For example, suppose the current reconstruction contains a straight line edge and that $\beta_1 = 1$, $\beta_2 = \beta_1/\sqrt{2}$ and $\sigma^2 = 1.0$. A white pixel which lies on this edge will change to black if its record is greater than 2.5. If this is not acceptable then the values of the parameters must be changed accordingly. This approach for the choice of these parameters is shown in more detail in Ripley (1986).

We refer to the part of the region which does not include pixels on the edges of areas of colour as the acceptable space for informative pixels for that particular reconstruction.

For a given reconstruction and value of k the size of the largest grouping is calculated. In fact the sizes of all groupings are calculated as these might be needed in the event of a tie. For the image in Figure 6.4 the sizes of the groupings were found and are shown in Table 6.1.

To test for significance a Monte Carlo simulation technique is used. Within the accepted space $N_{ir}(k)$ pixels are chosen at random without repetition, where $N_{ir}(k)$ is the number of informative residuals for the particular case. These pixels are marked and the sizes of the groupings are calculated in the same way as for

size of grouping	number of occurrences
1	293
2	21
3	11
4	4
5	0
6	1

Table 6.1; size of groupings for Figure 6.4.

Figure 6.4. If the random allocation contains a grouping which is bigger than the largest of the groupings found for the reconstruction then it is recorded as being more grouped. If it is smaller then it is recorded as being less grouped. If there is a tie then the second largest groupings are compared and so on, until a decision is made. This procedure is repeated for N random allocations and the number of times that more grouping is found is recorded. Under the null hypothesis the positioning of the informative residuals is uniformly random within the accepted space, so may be compared with the generated sample to give an indication of the significance of the grouping.

The choice of k is important. When k is small the number of informative residuals is large and so the test will be good at detecting the obliteration of large areas of colour. When the area that has been obliterated is small, k must be chosen to be large to increase the chances of getting a significant result from the grouping of the informative residuals which may lie in its place. Since the sizes of any obliterations (which may or may not exist) are not known, we test using several different values of k. We test at the 1% level of significance for k= 0, 1, 2, 3 and 4. If a significant result is obtained at any of the 5 levels then we have a result which is significant at at least the 5% level. In fact the test is slightly conservative because of the way that the five tests are combined.

6.3 Results

For the example in Figure 6.2 a test is carried out for 5 values of k. The results are shown in Table 6.2.

k	N	largest grouping	significance
0	2652	56	0.1 %
1	1151	36	0.1 %
2	390	6	0.4 %
3	93	3	1.6 %
4	25	2	13.7 %

Table 6.2. Results for Figure 6.2; $\sigma^2=0.3$; $\beta_1=0.8$; $\beta_2=\beta_1/\sqrt{2}$.

We now introduce another example as shown in Figure 6.9. The true value at each pixel is perturbed by Gaussian noise with mean 0 and variance 0.5. The exact MAP reconstruction is obtained for $\beta_1=0.9$ and $\beta_2=\beta_1/\sqrt{2}$ and this is shown in Figure 6.10. Figures 6.11-6.14 show the plots for k= 0, 1, 2 and 3 respectively. The results of the tests for these plots are shown in Table 6.3.

k	N	largest grouping	significance
0	3736	90	0.5 %
1	1342	21	0.2 %
2	338	10	0.1 %
3	55	3	0.2 %
4	6	1	-

Table 6.3. Results for Figure 6.10; $\sigma^2=0.5$; $\beta_1=0.9$; $\beta_2=\beta_1/\sqrt{2}$.

The results for both examples indicate that there is evidence to reject the null hypothesis in favour of the alternative, that the record has been oversmoothed, resulting in the obliteration of areas of colour. Using the same examples we reduce the values of the smoothing parameters used in reconstruction and test for oversmoothing in the same way.

Figure 6.15 shows the exact MAP reconstruction obtained using $\beta_1=0.6$ and $\beta_2=\beta_1/\sqrt{2}$ in the first example. The results of the tests for this reconstruction are shown in Table 6.4 and we can see that there is no evidence to reject the null

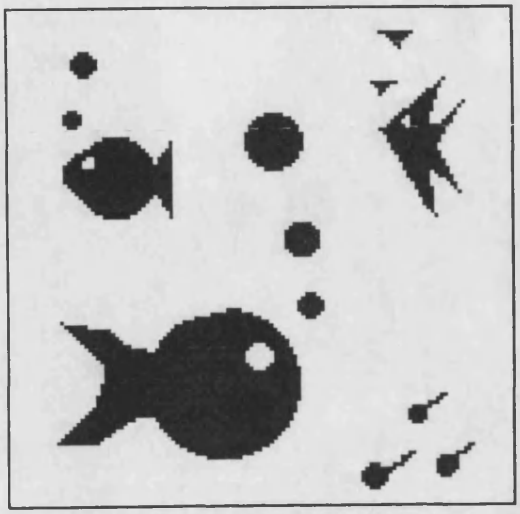


Figure 6.9
The true scene

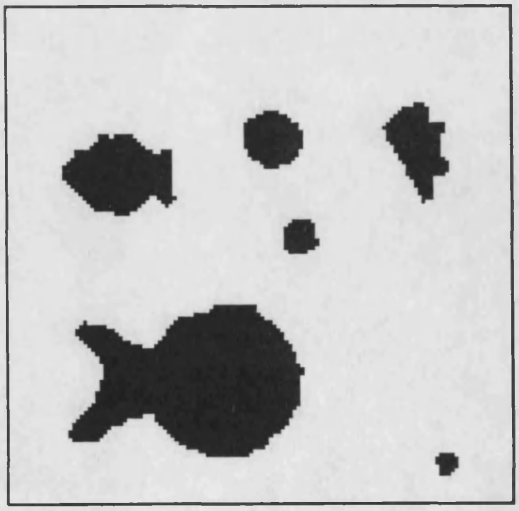


Figure 6.10
Reconstruction technique:
exact MAP
Parameters: $\beta_1 = 0.9$
$\beta_2 = \beta_1/\sqrt{2}$

Figure 6.11
Reconstruction Parameters:
Technique: exact MAP
$\beta_1 = 0.8$
$\beta_2 = \beta_1/\sqrt{2}$
Residual Parameters:
$k=0$
3736 informative residuals

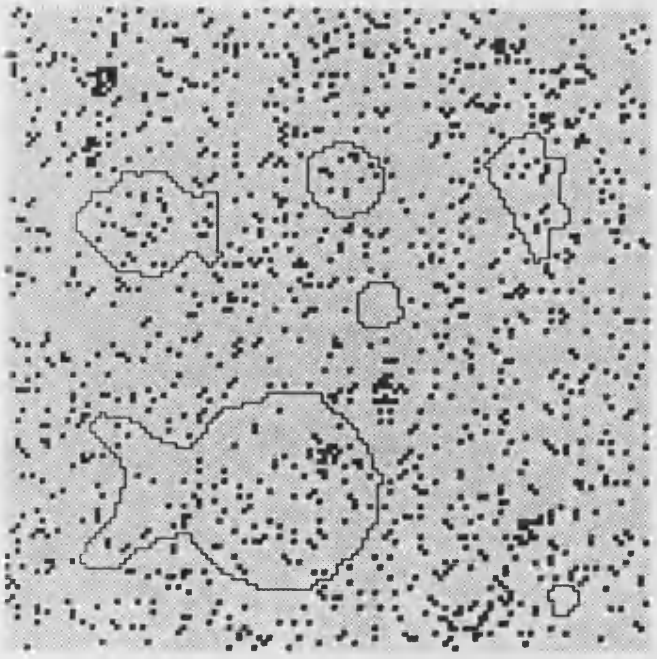


Figure 6.12
Reconstruction Parameters:
Technique: exact MAP
$\beta_1 = 0.8$
$\beta_2 = \beta_1/\sqrt{2}$
Residual Parameters:
$k=1$
1342 informative residuals

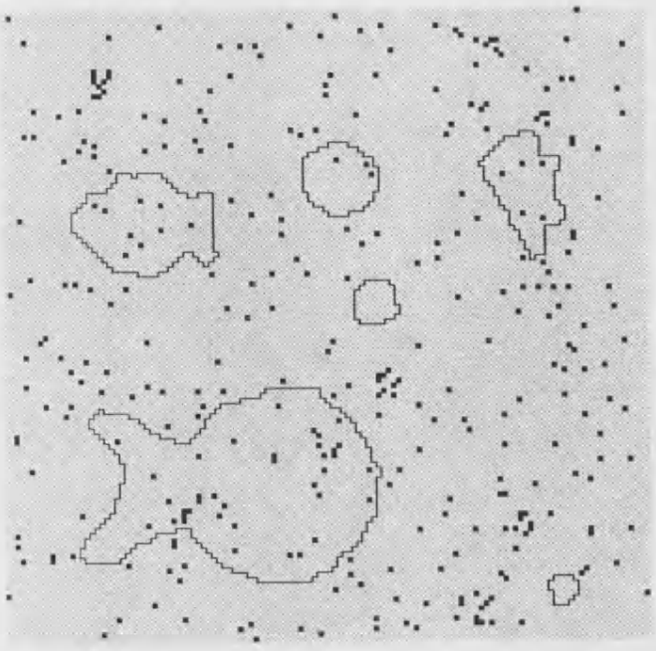


Figure 6.13
Reconstruction Parameters:
Technique: exact MAP
$\beta_1 = 0.8$
$\beta_2 = \beta_1/\sqrt{2}$

Residual Parameters:
$k=2$
338 informative residuals

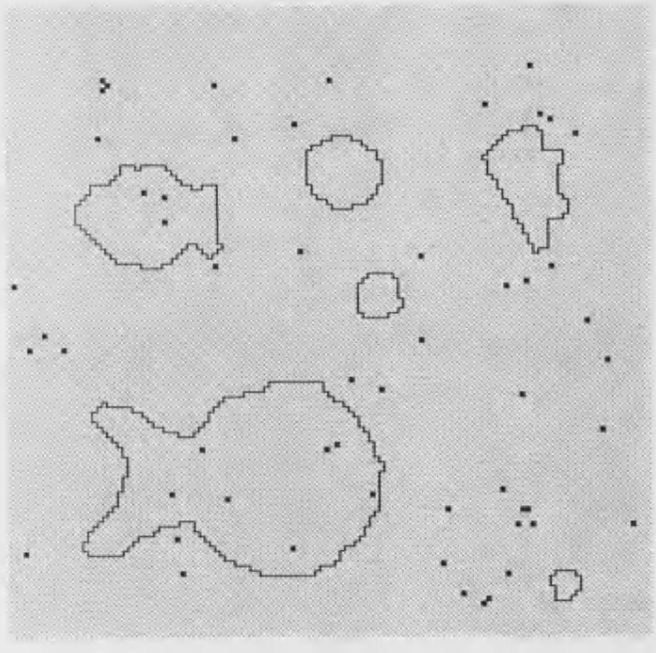


Figure 6.14
Reconstruction Parameters:
Technique: exact MAP
$\beta_1 = 0.8$
$\beta_2 = \beta_1/\sqrt{2}$

Residual Parameters:
$k=3$
55 informative residuals

hypothesis.

k	N	largest grouping	significance
0	2441	19	67.3 %
1	1081	7	65.1 %
2	307	4	13.2 %
3	51	1	-
4	1	1	-

Table 6.4. Results for Figure 6.15; $\sigma^2=0.3$; $\beta_1=0.6$; $\beta_2=\beta_1/\sqrt{2}$.

For the second example we obtain the exact MAP reconstruction shown in Figure 6.16 using $\beta_1=0.6$ and $\beta_2=\beta_1/\sqrt{2}$. There is still evidence of oversmoothing and this is shown in Table 6.5. The null hypothesis is rejected and the alternative is accepted.

k	N	largest grouping	significance
0	3567	90	0.3 %
1	1241	12	1.7 %
2	286	4	7.6 %
3	38	2	30.9 %
4	0	0	-

Table 6.5. Results for Figure 6.16; $\sigma^2=0.5$; $\beta_1=0.6$; $\beta_2=\beta_1/\sqrt{2}$.

This is chiefly due to the obliteration of one of the tadpoles during reconstruction, as can be seen from the residual plot for $k=0$, shown in Figure 6.17. Figure 6.18 shows the exact MAP reconstruction for this record, obtained using $\beta_1=0.55$ and $\beta_2=\beta_1/\sqrt{2}$. We can see that an area of black has appeared which represents an attempt to reconstruct the third tadpole. The test does not reject the null hypothesis for this reconstruction.

6.4 Concluding remarks

We have recorded the size of the largest grouping of informative residuals and compared this with the largest groupings of random allocations. This measure has been used to determine whether more or less grouping occurs. Clearly there

Figure 6.15
Reconstruction technique: exact MAP
Parameters: $\beta_1 = 0.6$
$\beta_2 = \beta_1/\sqrt{2}$

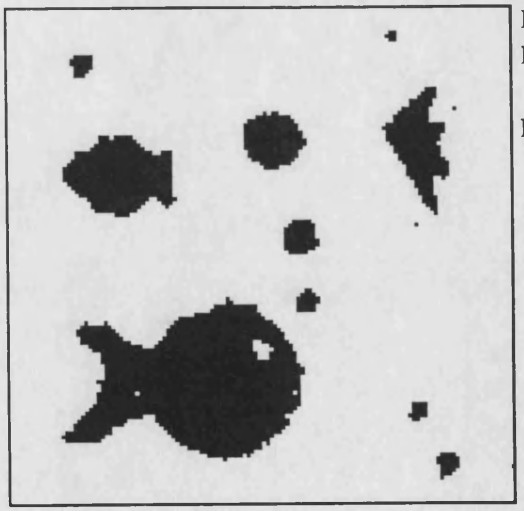


Figure 6.16
Reconstruction technique: exact MAP
Parameters: $\beta_1 = 0.6$
$\beta_2 = \beta_1/\sqrt{2}$

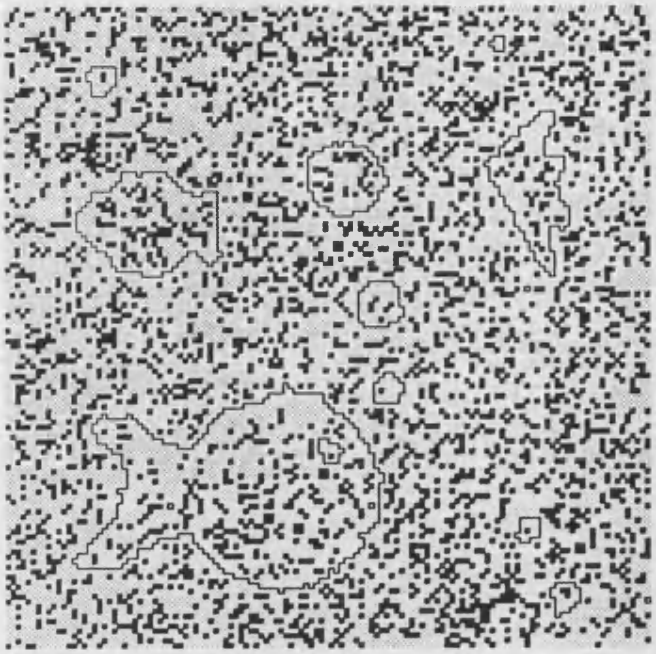


Figure 6.17
Reconstruction Parameters:
Technique: exact MAP
$\beta_1 = 0.6$
$\beta_2 = \beta_1/\sqrt{2}$
Residual Parameters:
$k = 2$
3597 informative residuals

Figure 6.18
Reconstruction technique:
exact MAP
Parameters: $\beta_1 = 0.55$
$\beta_2 = \beta_1/\sqrt{2}$

are many other ways in which grouping could be measured in this context. For example the average size of the m largest groupings could be measured or a weighted average used. The system that we have adopted is intuitively sound and, as is demonstrated by the results, appears to work well.

We have not discussed the problems of finding an initial estimate for β, nor have we given a strategy for the reduction of the β parameter when oversmoothing is detected. However we have provided a procedure which, for a given reconstruction, provides a graphical aid and a statistical test which helps to determine whether oversmoothing has occurred.

Chapter 7: An approximate grey level MAP technique

For any reconstruction problem involving more than two colours there is currently no recognised technique for finding the exact MAP reconstruction, other than by considering all possible configurations. In general, simulated annealing or ICM are used although the quality of the reconstruction will depend on the nature of the true scene, the level of noise and the suitability of the chosen prior. In this chapter we combine the approach used by Derin *et al.* (1985) which divides the problem into several problems, each involving only two colours, with the exact MAP reconstruction technique proposed by Grieg, Porteous and Seheult (1989) for binary images. These methods have been described in Section 3.5 and 3.4 respectively.

7.1 The method

The basis of the method is to identify parts of the region which contain only two colours and to obtain a reconstruction on this subregion using the exact MAP technique for binary reconstruction. Combining several subregion estimates provides a multi-level reconstruction on the whole region. The method is capable of handling as many grey levels as are required but we shall describe it for the simple case of only three underlying levels, 0, 1 and 2. We obtain a reconstruction on the whole region S using the exact MAP technique where each pixel is assigned a colouring 0 or 1. We denote this reconstruction by $x_{S(0,1)}$. Although this reconstruction is not applicable for a three level problem it is used as an intermediate reconstruction. We have assumed that the true 3 colour scene is the realisation of some form of Markov random field (MRF). If all the 2's in the true scene are changed to 1's then the resulting scene contains only two colours. A binary MRF is used to model this scene and the reconstruction $x_{S(0,1)}$ that has been obtained may be regarded as an estimate of this scene. Consider all the pixels in the original true scene which are coloured as 1's and 2's and denote this subregion by $S_{1,2}$. We assume that the configuration of 1's and 2's in this subregion is a realisation of a binary MRF and obtain an estimate of this subregion in the following way. Form the set of pixels S_1 which are coloured 1 in the reconstruction $x_{S(0,1)}$ and obtain the reconstruction $x_{S_1(1,2)}$ on the subregion S_1 where each pixel may be coloured 1 or 2. Combining this reconstruction with the pixels which were coloured 0 in $x_{S(0,1)}$ provides a three level reconstruction on

the whole region S. This is unlikely to be the exact MAP reconstruction for this problem but may provide a useful reconstruction. The order in which the levels are considered is referred to as the *splitting schedule.* In the above case the splitting schedule was to find $x_{S(0,1)}$ and then $x_{S_1(1,2)}$ but a three level reconstruction could be obtained equally easily by finding $x_{S(1,2)}$ and then $x_{S_1(0,1)}$. Additional splits may be added in the final stages of reconstruction. For example the schedule $x_{S(0,1)}$, $x_{S_1(1,2)}$ may be followed by $x_{S_{0,1}(0,1)}$, $x_{S_{1,2}(1,2)}$ where $S_{l,m}$ denotes the region of pixels coloured l or m. The third and fourth splits of this schedule may be regarded as confirmation splits where hopefully very few changes take place.

We shall illustrate the use of this technique with a simple four level example in which each pixel is coloured 0, 1, 2 or 3. Figure 7.1 shows the true scene. Independent Gaussian noise with mean 0 and variance 1.0 is added to the true value at each pixel and the closest mean classifier is shown in Figure 7.2, where over 47% of the pixels are misclassified. Figure 7.3 shows the reconstruction $x_{S(1,2)}$, obtained using $\beta_1=0.8$ and $\beta_2=\beta_1/\sqrt{2}$, where each pixel takes the value 1 or 2. The sets S_1 and S_2 are then formed and, using the same parameters, the reconstructions $x_{S_1(0,1)}$ and $x_{S_2(2,3)}$ are found. These reconstructions may then be combined and Figure 7.4 shows the resulting four colour reconstruction. The time required to obtain the estimate is less than twice that required to find the exact MAP estimate for a black and white scene under similar conditions.

The prior which is used in this example is

$$p(x)\propto\exp[-\{\beta_1 Z_1(x)+\beta_2 Z_2(x)\}] \tag{7.1}$$

where $Z_1(x)$ and $Z_2(x)$ are the number of first and second order discrepant pairs in the scene x. It is interesting to compare the reconstruction that we have obtained with Figure 7.5, which is the ICM reconstruction for $\beta_1=1.3$, $\beta_2=\beta_1/\sqrt{2}$ and the prior given by (7.1). The β parameters that were used gave the "best" results for the method but it is clear that the approximate MAP reconstruction is superior.

Because the final reconstruction is not necessarily the MAP reconstruction, it is not unique and if different splitting schedules are used then different reconstructions may be obtained. In the example that we have shown the triangular shape on the right hand side of the image in Figure 7.1 has been eroded

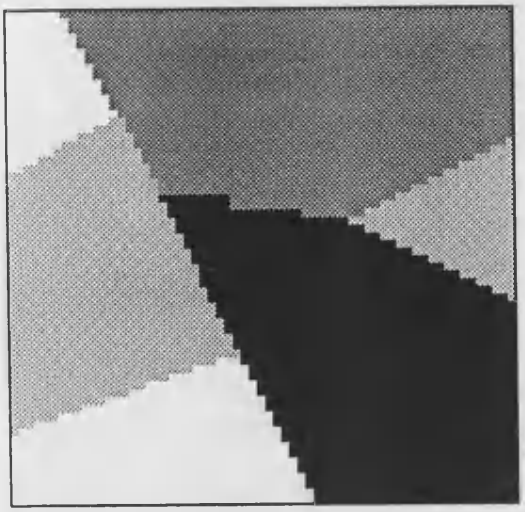


Figure 7.1

The true scene
Size 64 × 64
Levels: 0, 1, 2, 3.

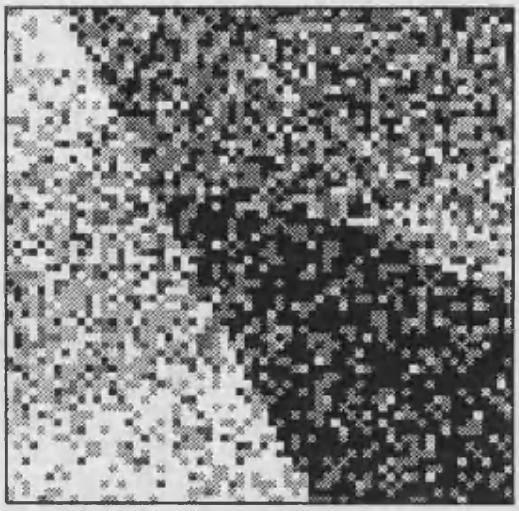


Figure 7.2

The closest mean classifier
Number of pixels misclassified : 1946

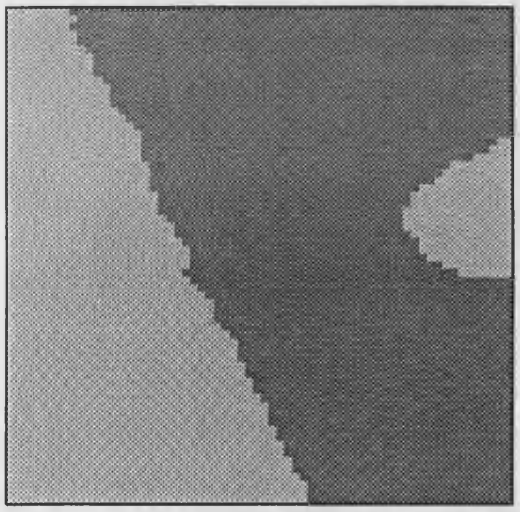


Figure 7.3

Reconstruction technique: Exact MAP,
for levels 1 and 2
Parameters: $\beta_1 = 0.8$
$\beta_2 = \beta_1/\sqrt{2}$

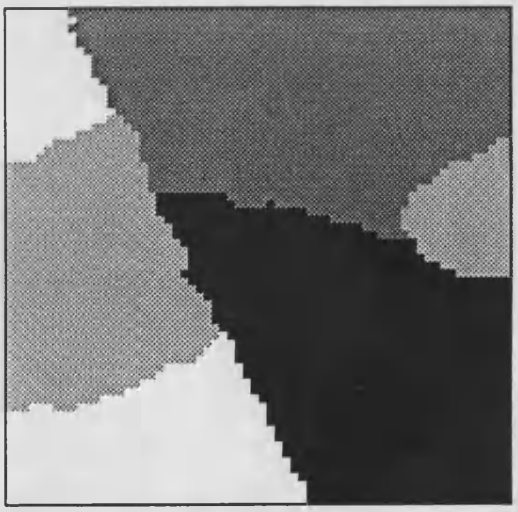


Figure 7.4

Reconstruction technique: Approximate MAP,
see text
Parameters: $\beta_1 = 0.8$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: Figure 7.3
Number of misclassified pixels: 101

in Figure 7.3. This is partly due to the 2's and 3's being grouped together against the 1's at this stage of the process. One way in which the effect of this may be limited is to include an extra split at the end of the reconstruction process in which the set $S_{1,2}$ of pixels coloured 1 or 2 is formed and the reconstruction $x_{S_{1,2}(1,2)}$ obtained. This step may seem unnecessary but Figure 7.6 shows the reconstruction obtained when this extra step is included and we can see that it has helped, reducing the misclassification rate to less than 2%. Notice that pixels coloured 0 or 3 in Figure 7.4 have not been affected by this extra step, this is because they are not included in the subregion $S_{1,2}$ being reconstructed in the final split.

7.2 The formulation of the subregion problem

We now give details of how exact MAP estimates may be obtained on subregions of the image. This involves showing that the maximisation problem over a subregion A of the region S may be expressed in such a way that the network flow formulation may be used to find the estimate. Consider a grey level region S in which pixels are coloured one of l levels $0,\ldots,l-1$. The pixel colourings are perturbed by some noise, the record at each pixel having the same known conditional density function $f(y_i \mid x_i)$, dependent only on x_i. The prior is given by

$$p(x) \propto \exp\{-\frac{\beta}{2}\sum_{i=1}^{n}\sum_{j\in\partial_i}\Phi(x_i,x_j)\}$$

where Φ is some function of x_i and x_j. For the first part of this section we assume that $\Phi(x_i,x_j)=I(x_i \neq x_j)$. (Recall $I(x_i \neq x_j)=0$ if $x_i=x_j$ and 1 otherwise.) We might wish to find the MAP estimate of the scene, i.e. the value of x which maximises

$$l(y \mid x)p(x)$$

or equivalently

$$\ln l(y \mid x) - \frac{\beta}{2}\sum_{i=1}^{n}\sum_{j\in\partial_i}\Phi(x_i,x_j). \qquad (7.2)$$

Suppose A is a subregion of S and that we wish to find the MAP estimate on the region A such that pixels in A only take the value a or b for some positive integers a,b where $a<b<l$. Suppose also that the set $S\backslash A$ of pixels which are not

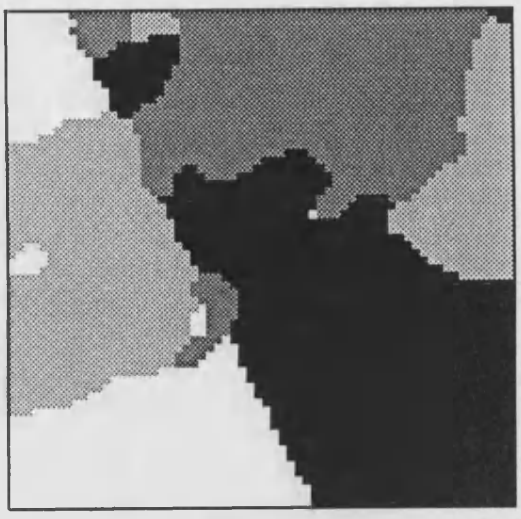


Figure 7.5
Reconstruction technique: ICM
Number of pixels misclassified : 1946
Parameters: $\beta_1 = 1.3$
$\beta_2 = \beta_1/\sqrt{2}$
Initial estimate: Closest mean classifier
Number of misclassified pixels: 517

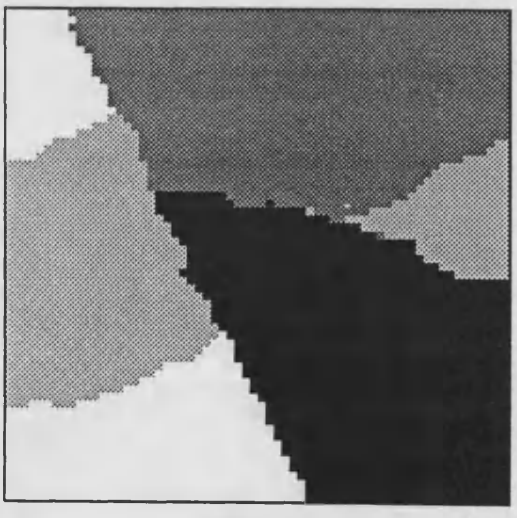


Figure 7.6
Reconstruction technique: Approximate MAP, see text
Parameters: $\beta_1 = 0.8$
$\beta_2 = \beta_1/\sqrt{2}$
Number of misclassified pixels: 83

in the set A does not contain any pixels which are coloured a or b. We denote a colouring of the region A in which the pixels are coloured a or b by $x_{A(a,b)}$. We denote the exact MAP estimate by $\hat{x}_{A(a,b)}$ and note that it is the colouring of A which maximises (7.2) with all other colourings fixed. We may rewrite (7.2) as

$$\sum_{i\in A}\ln l(y_i\,|\,x_i)+\sum_{i\in S\setminus A}\ln l(y_i\,|\,x_i)-\frac{\beta}{2}\sum_{i\in A}\sum_{j\in(\partial_i\cap A)}\Phi(x_i,x_j)$$

$$-\beta\sum_{i\in A}\sum_{j\in(\partial_i\cap S\setminus A)}\Phi(x_i,x_j)-\frac{\beta}{2}\sum_{i\in S\setminus A}\sum_{j\in(\partial_i\cap S\setminus A)}\Phi(x_i,x_j). \qquad (7.3)$$

Observe that the second and fifth terms of (7.3) remain fixed through all colourings of A and so may be ignored at this stage of the process. The fourth term of (7.3) can be regarded as a boundary term. This term also remains constant since $\Phi(x_i,x_j)=I(x_i\neq x_j)$, $S\setminus A$ contains no pixels coloured a or b and A contains only pixels coloured a or b. So we require the value of $x_{A(a,b)}$ which maximises

$$\sum_{i\in A}\ln l(y_i\,|\,x_i)-\frac{\beta}{2}\sum_{i\in A}\sum_{j\in(\partial_i\cap A)}\Phi(x_i,x_j)$$

$$=\sum_{i\in A}\ln f(y_i\,|\,x_i=b)^{I(x_i=b)}f(y_i\,|\,x_i=a)^{1-I(x_i=b)}-\frac{\beta}{2}\sum_{i\in A}\sum_{j\in(\partial_i\cap A)}\Phi(x_i,x_j)$$

$$=\sum_{i\in A}I(x_i=b)\lambda_i-\frac{\beta}{2}\sum_{i\in A}\sum_{j\in(\partial_i\cap A)}\Phi(x_i,x_j) \qquad (7.4)$$

where $\lambda_i=\ln[f(y_i\,|\,x_i=b)/f(y_i\,|\,x_i=a)]$. Replacing $\Phi(x_i,x_j)$ by $I(x_i\neq x_j)$ in (7.4) the expression is now in such a form that the network flow formulation may be used to find $\hat{x}_{A(a,b)}$. Details of how this may be done are shown in Section 3.4.

Dealing with more intricate priors.

We have shown the formulation of the problem for the simple case where $\Phi(x_i,x_j)=I(x_i\neq x_j)$. We now consider the maximisation where the function $\Phi(x_i,x_j)$ may take any form, although we shall assume that $\Phi(x_i,x_j)=0$ if $x_i=x_j$. Using (7.4) we may rewrite (7.3), and we require the estimate $x_{A(a,b)}$ which maximises

$$\sum_{i\in A}I(x_i=b)\lambda_i+\sum_{i\in S\setminus A}\ln l(y_i\,|\,x_i)-\frac{\beta}{2}\sum_{i\in A}\sum_{j\in(\partial_i\cap A)}\Phi(x_i,x_j)$$

$$-\beta\sum_{i\in A}\sum_{j\in(\partial_i\cap S\setminus A)}\Phi(x_i,x_j)-\frac{\beta}{2}\sum_{i\in S\setminus A}\sum_{j\in(\partial_i\cap S\setminus A)}\Phi(x_i,x_j)$$

where $\lambda_i=\ln[f(y_i\,|\,x_i=b)/f(y_i\,|\,x_i=a)]$. Again the second and fifth terms of the

expression may be ignored. However the fourth term must now be included in the maximisation since it may be affected by the colouring of the region A. We may rewrite the fourth term as

$$-\beta\sum_{i\in A}\sum_{j\in(\partial_i\cap S\setminus A)} I(x_i=b)\left[\Phi(b,x_j)-\Phi(a,x_j)\right]-\beta\sum_{i\in A}\sum_{j\in(\partial_i\cap S\setminus A)}\Phi(a,x_j). \quad (7.5)$$

The second term of (7.5) remains constant over all colourings of A and may therefore be ignored. It follows that the map estimate $\hat{x}_{A(a,b)}$ is the value of $x_{A(a,b)}$ which maximises

$$\sum_{i\in A} I(x_i=b)(\lambda_i-\beta k_i)-\frac{\beta}{2}\sum_{i\in A}\sum_{j\in(\partial_i\cap A)}\Phi(x_i,x_j)$$

where

$$\lambda_i=\ln[f(y_i\,|\,x_i=b)/f(y_i\,|\,x_i=a)]$$

and

$$k_i=\sum_{j\in(\partial_i\cap S\setminus A)}[\Phi(b,x_j)-\Phi(a,x_j)].$$

We may rewrite this as

$$\sum_{i\in A} I(x_i=b)(\lambda_i-\beta k_i)-\frac{\beta}{2}\Phi(a,b)\sum_{i\in A}\sum_{j\in(\partial_i\cap A)} I(x_i\neq x_j)$$

and this form may be used to construct the network flow formulation.

7.3 Concluding Remarks

Although the method that we have described does not provide the exact grey level MAP estimate, the reconstruction produced may be of good quality. The example that we have shown illustrates that where there are few grey levels, satisfactory reconstructions may be obtained, even when the noise level is high. We have implemented a version of the program which is capable of handling any form of the function $\Phi(x_i,x_j)$ in the prior and have found that the quality of reconstructions obtained depends on the suitability of the splitting schedule as well as the prior, particularly when the number of grey levels is high. At the very least the method may be used as a final tidying up step for reconstructions obtained using other methods such as ICM or simulated annealing. Subregions containing adjacent colours in the reconstruction can be identified and isolated and the exact two colour MAP method applied.

It is interesting to note that the final stages of this method involves reconstruction on subregions which contain only two levels. A natural extension of the method would be to consider these subregions in turn, and use the subpixel refinement process which was described in Section 5.5 to get a subpixel grey level reconstruction on the whole region.

References

Besag, J. E. (1972). On the statistical analysis of nearest-neighbour systems. *Proc. Eur. meeting of statisticians* . Budapest, 1972.

Besag, J. E. (1974). Spatial interaction and the statistical analysis of lattice systems. *J. R. Statist. Soc. B*, **36**, 192-326.

Besag, J. E. (1986). On the statistical analysis of dirty pictures. *J. R. Statist. Soc. B*, **48**, 259-302.

Cross, G. R. and Jain, A. K. (1983). Markov random field texture models. *IEEE Trans. Pattern Anal. Machine Intel.*, vol. PAMI-**5**, 25-39.

Derin, H., Elliot, H., Cristi, R. and Geman, D. (1984). Bayes smoothing algorithms for segmentation of binary images modeled by Markov random fields. *IEEE Trans. Pattern Anal. Machine Intel.*, vol. PAMI-**6**, No. 6, 707-720.

Derin, H., Elliot, H., Soucy, R. M. and Kuang, J. (1985). Segmentation of SAR imagery using Gibbs distribution models. *Proc. of Eleventh Int. Symp. Machine Proc. Remotely Sensed Data.* West Lafayette, Indiana, USA.

Ford, L. R. and Fulkerson, D. R. (1962). *Flows in Networks.* Princeton University Press.

Frigessi, A. and Piccioni, M (1988). Parameter estimation for 2-dimensional Ising fields corrupted by noise. *Spatial Statistics and Imaging, IMS Lecture notes.* Maine, USA. 1988.

Geman, D. (1985). Bayesian image analysis by adaptive annealing. *Dig. 1985 Int. Geosc. and Rem. Sens. Symp.* Amherst, MA, USA. 1985.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intel.*, vol. PAMI-**6**, No. 6, 721-741.

Geman, S. and Graffigne, C. (1986). Markov random field image models and their application to computer vision. *Proc. Int. Cong. Math.*, Berk., Calif., USA., 1986, 1496-1517.

Geman, S. and McClure, D. E. (1985). Bayesian image analysis: An application to single photon emission tomography. *Statistical Computing Section, Proc. Amer. Statist. Assc.*, 1985, 12-18.

Gidas, B. (1985a). Non-stationary Markov chains and convergence of the annealing algorithm. *Jour. Statist. Phys.*, **39**, 1985, 73-131.

Gidas, B. (1986). Consistency of maximum likelihood and pseudo-likelihood estimators for gibbs distributions. *Proc. Workshop on Stoch. Diff. Sys. with Appl. in Elec./Comp. Eng., Control Theory and Oper. Res.*, IMA, Univ. of Minnesota, 1986.

Gidas, B. (1985b). "Parameter estimation for Gibbs distributions." Div. Appl. Math., Brown University (unpublished).

Gidas, B. (1989). A renormalization group approach to image processing problems. *IEEE Trans. Pattern Anal. Machine Intel., to appear.*

Grenander, U. (1983). "Tutorial in pattern theory", Div. Appl. Math., Brown University (unpublished).

Grenander, U. and Osborn, B. E. (1983). "Parameter estimation in pattern theory", Div. Appl. Math., Brown University (unpublished).

Grenander, U. (1985). "Advances in pattern theory", Div. Appl. Math., Brown University (unpublished).

Grieg, D. M., Porteous, B. T. and Seheult, A. H. (1989). Exact Maximum *a Posteriori* estimation for binary images. *J. R. Statist. Soc. B,* **51** (1989), No. 2., 271-279.

Hall, P. and Titterington, D. M. (1986). On some smoothing techniques used in image restoration. *J. R. Statist. Soc. B,* **48** (1986), No. 3., 330-343.

Huertas, A. and Medioni, G. (1986). Detection of intensity changes with subpixel accuracy using Laplacian-Gaussian masks. *IEEE Trans. Pattern Anal. Machine Intel.*, vol. PAMI-**8**, No. 5, 651-663.

Ising, E. (1925). *Zeitschrift Physik,* Vol. 31, p253.

Jennison, C. (1986). Contribution to discussion of Besag (1986). *J. R. Statist. Soc. B,* **148**, 288-289.

Kashyap, K. and Chellappa, R. (1983). Estimation and choice of neighbors in spatial-interaction models of images. *IEEE Trans. Inf. Theory, vol.* **IT-29**, 60-72.

Lakshmanan, S. and Derin, H (1989). Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing. *IEEE*

Trans. Pattern Anal. Machine Intel., vol. PAMI-**11**, 799-813.

Pars, L. A. (1962). *Introduction to the calculus of variations*. Heinemann, London.

Ripley, B, D. (1986). Statistics, images and pattern recognition. *Canad. J. Statist.* **14**, 83-111.

Ripley, B, D. (1988a). *Statistical Inference for Spatial Processes*. Cambridge University Press, Cambridge.

Ripley, B, D. (1988b). The use of spatial models as image priors. *Spatial Statistics and Imaging, IMS Lecture notes*. Maine, USA. 1988.

Silverman, B. W., Jennison, C. and Brown, T. C., (1989). Edge process models for regular and irregular pixels. *to appear.*

Stander, J. (1989). Research report. Department of Math. Sci., Univ. of Bath. (unpublished).

Switzer, P. (1983). Some spatial statistics for the interpretation of satellite data. *Bull. Int. Statist. Inst.* **50**, 962-971.

Switzer, P. and Venetoulias, A. (1985). Spatial classification error rates related to pixel size. Tech. Rep. No. 11. August 1985. Dep. of Statist. Stanford Univ. USA.